



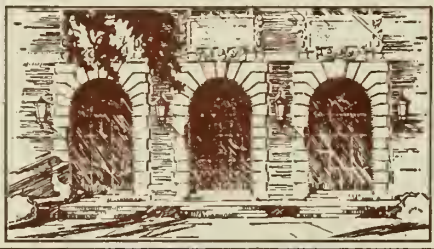
LIBRARY OF THE  
UNIVERSITY OF ILLINOIS  
AT URBANA-CHAMPAIGN

510.84

Il6r

no. 800 - 805

cop. 2







Digitized by the Internet Archive  
in 2013

<http://archive.org/details/computationoffou802koch>

010.01  
Ill  
no. 802  
cop 2.

Math

UIUCDCS-R-76-802

COMPUTATION OF FOUR COLOR IRREDUCIBILITY

by

John Allen Koch

May 1976



The Library of the

JUN 28 1976

University of Illinois  
at Urbana-Champaign

DEPARTMENT OF COMPUTER SCIENCE  
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN · URBANA, ILLINOIS



UIUCDCS-R-76-802

COMPUTATION OF FOUR COLOR IRREDUCIBILITY

by

John Allen Koch

May 1976

Department of Computer Science  
University of Illinois at Urbana-Champaign  
Urbana, Illinois 61801

This work was supported in part by the Department of Computer Science and in part by the National Science Foundation under Grant MCS 73-03408, and was submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science, 1976.



## ACKNOWLEDGMENTS

The author wishes to express his gratitude to his thesis advisor, Professor Kenneth Appel, and to his Committee Chairman, Professor C. L. Liu, for their many suggestions and continuous encouragement.

Thanks are also in order to my office mates, Jim Bitner and Brian Hansche, for many enlightening discussions. The consultants of the Computing Services Office are also to be thanked for their assistance in debugging and system interface. The speedy and accurate typing of Connie Slovak is greatly appreciated.

Finally, this thesis is dedicated to my wife, Mimi, whose support and encouragement enabled this thesis to be completed.



## TABLE OF CONTENTS

| CHAPTER                                      | Page |
|----------------------------------------------|------|
| 1. INTRODUCTION AND HISTORY.....             | 1    |
| 2. DEFINITIONS.....                          | 6    |
| 3. DIRECT AND INDIRECT COLORATIONS.....      | 25   |
| 4. REDUCER GENERATION.....                   | 40   |
| 5. IMPLEMENTATION OF REDUCER GENERATION..... | 58   |
| 6. C-REDUCTION.....                          | 72   |
| 7. PROGRAM OUTPUT.....                       | 80   |
| 8. CONCLUSIONS AND EXTENSIONS.....           | 88   |
| LIST OF REFERENCES.....                      | 91   |
| APPENDIX A.....                              | 93   |
| APPENDIX B.....                              | 103  |
| APPENDIX C.....                              | 106  |
| APPENDIX D.....                              | 108  |
| APPENDIX E.....                              | 112  |
| VITA.....                                    | 114  |



## CHAPTER 1

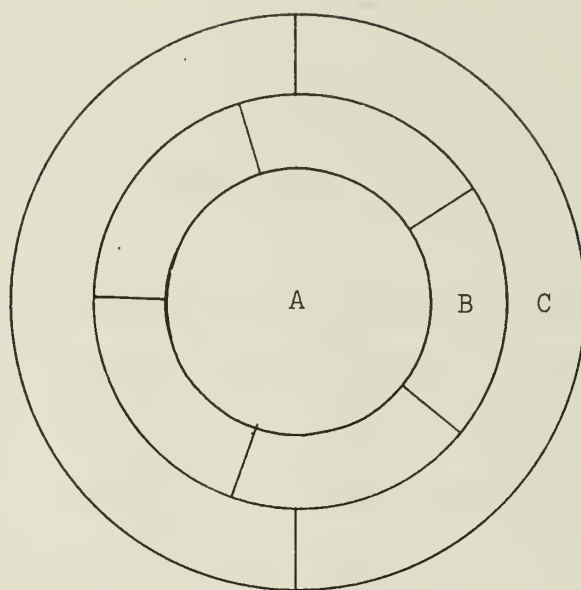
### INTRODUCTION AND HISTORY

The Four Color Conjecture (4CC) has remained a conjecture for over 100 years despite numerous attempts to prove or disprove it. The 4CC is intriguing because it can be simply stated but has been so difficult to prove. The Conjecture is concerned with coloring the countries (regions) of a planar map. A planar map may be thought of as a graph that can be embedded in the plane. A country of the map must be a region of the graph. Two countries, say A and B of Figure 1.1(a), are called adjacent when they share a common border that is not just a single point. A coloring of the map is an assignment of a color to each country.

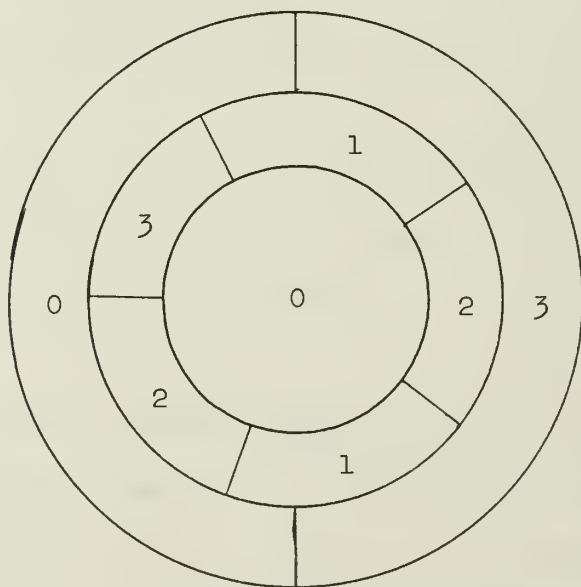
Four Color Conjecture: The countries of every planar map can be colored using a maximum of 4 colors so that no two countries that are adjacent have the same color.

The dual formulation of the 4CC in graph theoretic terms is used in this thesis. (All terms used in Chapter 1 are defined in Chapter 2.) The conjecture now becomes:

Four Color Conjecture: The vertices of every planar graph can be colored using a maximum of 4 colors so that no two adjacent vertices have the same color.



(a) Countries of a map



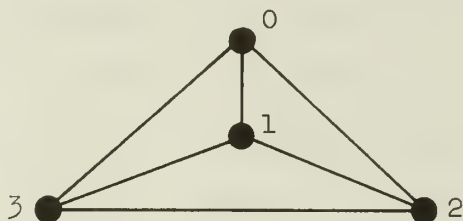
(b) A 4-coloring of a map

Figure 1.1 Planar maps

The history of the 4CC is interesting and varied. The first known statement of the 4CC was made in about 1856 by Frances Guthrie [10]. The first erroneous proof of the 4CC was published by A. B. Kempe [15] in 1879. His proof contained the first configurations that were shown to be reducible. A configuration may be thought of as a connected, completely triangulated graph surrounded by a circuit of vertices. Informally, a configuration is reducible if it cannot be a subgraph of a minimal counterexample to the 4CC. (This is only an informal definition since the reducibility of a configuration might depend on the truth of the 4CC.) Kempe's proof used a set of configurations which was unavoidable in the sense that it could be shown that every planar graph had to contain at least one element of this set. By supposedly showing all the elements of his unavoidable set were reducible, Kempe thought he had proven that a counterexample to the 4CC could not exist. However, the demonstration of reducibility for one configuration of Kempe's unavoidable set was incorrect and this fact appeared in P. J. Heawood's rebuttal [13] in 1890. Heawood was able to use Kempe's reduction techniques to prove that all planar graphs can be colored using 5 or fewer colors. In 1913, G. D. Birkhoff [8] showed that several configurations were reducible by using a technique, based on Kempe's reduction technique, that became known as C-reduction. P. Franklin [9] used C-reduction to find more reducible configurations and to prove that the 4CC was true for graphs with fewer than 26 vertices. By showing sets of reducible configurations which are unavoidable in all planar graphs with a restricted number of vertices, C. E. Winn [24], O. Ore and J. Stemple [19], W. Stromquist [21] increased this restricted number

until J. Mayer [17] has recently shown that all planar graphs with fewer than  $96$  vertices are 4-colorable.

It is interesting to note that the number of colors necessary and sufficient to color all graphs with genus higher than the genus of planar graphs has been exactly determined. For planar graphs, clearly there are some graphs that require more than four colors:



Thus, 4 colors are necessary and 5 colors are sufficient to color all planar graphs. For graphs of higher genus, G. Ringel and J. W. T. Youngs (see [20] for further references) were able to show that

$\left\lfloor \frac{7 + \sqrt{1+48n}}{2} \right\rfloor$  colors are sufficient to color all graphs of genus  $n > 0$ .

Since Birkhoff introduced C-reduction, many configurations have been shown to be reducible (see Tables I and II [1]). This thesis is concerned with the problem of exhaustively checking configurations for reducibility so that if a configuration is not shown C-reducible, one may assert its C-irreducibility. This exhaustive study of reducibility for all configurations of 15 or fewer vertices that have a ring size,  $n$ ,  $6 \leq n \leq 11$ , and  $m$  interior vertices,  $m \leq 15 - n$  has important implications for configurations with larger ring size ( $n > 11$ ) (discussed in Chapter 8). A configuration is C-irreducible when it is not C-reducible. (The reduction methods are described in Chapters 2 and 3.) Not much work has been done to show configurations to be irreducible [11]. The number of calculations needed to compute irreducibility requires the use of a computer on all but the smallest configurations.

Chapter 2 contains definitions of terms, methods of reduction and the reasons for examining configurations for reducibility. Chapters 3 and 6 concern implementation of the methods of reduction; while Chapters 4 and 5 detail the generation of potential reducers of configurations. Demonstration of irreducibility and extensions and conclusions of this work are discussed in the final two chapters, 7 and 8.

## CHAPTER 2

## DEFINITIONS

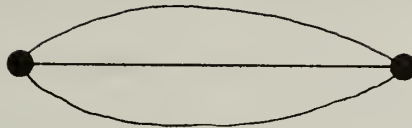
In this chapter, concepts and basic terminology of graph theory and reduction are presented. The graph theoretic terminology is basically the same as the terminology of Harary [12].

An abstract graph consists of a set of vertices,  $V$ , along with a specified set,  $E$ , of unordered pairs of distinct vertices of  $V$ . Since a set is a collection of distinct elements, each pair of vertices is allowed to appear only once in  $E$ . An edge is a pair of vertices,  $x = (u,v)$ , in  $E$  and the edge,  $x$ , joins vertex  $u$  and vertex  $v$ ; vertex  $u$  and edge  $x$  are said to be incident with each other. Two vertices that are joined by an edge are called adjacent vertices. Note that the definition of an abstract graph does not allow self-loops (an edge that joins a vertex to itself as in Figure 2.1(a)) or multiple edges (when two vertices are joined by more than a single edge as in Figure 2.1(b)).

An abstract graph is usually represented by a diagram which is referred to as a graph. In the graph of Figure 2.2, the vertices are the lettered points and an edge is a line that joins two adjacent vertices. The degree of a vertex is the number of edges incident upon that vertex. In Figure 2.2, vertex  $b$  has degree 4; vertex  $d$  has degree 1. A planar graph is a graph that can be drawn in the plane so that no edges intersect. Figure 2.2 shows a planar graph,  $G$ , and one method of drawing  $G$  in the plane. All graphs from now on are assumed to be planar. A

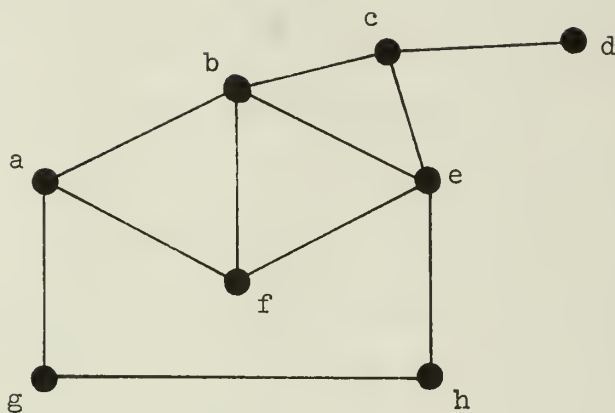


(a) Self-loop



(b) Multiple edges  
joining b and c

Figure 2.1 Subgraphs that are not allowed.



A connected, planar graph,  $G$ , with 8 vertices, 11 edges and 5 regions.

$$G = (V, E)$$

$$V = \{a, b, c, d, e, f, g, h\}$$

$$E = \{(a, b), (a, f), (a, g), (b, c), (b, e), (b, f), (c, d), (c, e), (e, f), (e, h), (g, h)\}$$

Figure 2.2 A planar connected graph.

region of a graph is an area of the plane that is bordered by edges of the graph and that contains no vertices or edges of the graph.

There are 5 regions in Figure 2.2. (The area outside the graph is also a region.) A subgraph of a graph  $G$  is itself a graph consisting of some vertices and edges of  $G$ .

The property that a graph is connected is developed in the following section. A walk of a graph,  $G$ , is a sequence of vertices,  $v_0, v_1, v_2, \dots, v_n$ , in which each sequential pair of vertices of the walk must be an edge of  $G$ . A path of  $G$  is a walk in which all the vertices of the walk are distinct. A walk is closed if  $v_0 = v_n$ . A closed walk is a circuit provided its  $n$  vertices are distinct and  $n \geq 3$ . A graph is connected if it is possible to find a path that joins any two vertices of the graph.

A proper coloring of a graph is the assignment of a color to each vertex of the graph so that no two adjacent vertices have the same color. From now on, a coloring of a graph is assumed to be a proper coloring. An  $n$ -coloring of a graph is a coloring of the graph that uses a maximum of  $n$  different colors.

All graphs that are considered in this thesis will be completely triangulated. A graph is completely triangulated if all regions of the graph are bordered by exactly 3 edges. This restriction can be imposed for the following reasons. If all completely triangulated planar graphs are 4-colorable, then all planar graphs are 4-colorable. Conversely, if a planar graph,  $G$ , is not 4-colorable, then a completely triangulated graph,  $G'$ , which may be obtained by adding edges (but no vertices) to  $G$ , cannot be 4-colorable.

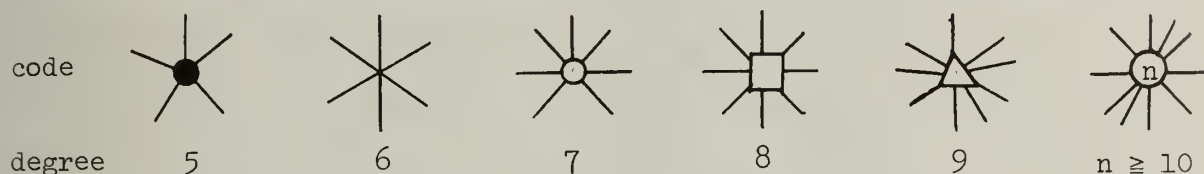
The transformation from planar maps to the dual formulation as planar graphs can be described as follows. Each country of the map is represented by a vertex in the graph. Two vertices are joined in the graph if their corresponding countries are adjacent in the map. A vertex is then assigned the color that is assigned to its country in the map.

Definitions that are peculiar to four color reducibility theory will now be presented. A minimal five chromatic graph is a non-4-colorable graph such that any graph with fewer vertices is 4-colorable. A minimal five chromatic graph may be thought of as a counterexample to the 4CC that has the smallest number of vertices of all counterexamples. (The existence of a counterexample to the 4CC is not presumed; a possible proof of the 4CC consists of proving that a minimal five chromatic graph cannot exist.)

An n-ring is a circuit of  $n$  vertices. An  $n$ -ring divides the plane into two parts, the interior and the exterior of the ring. Vertices and edges inside the  $n$ -ring are called interior vertices and interior edges. The vertices of the  $n$ -ring are ring vertices, and edges that join two consecutive ring vertices are ring edges. A diagonal is an interior edge that joins two ring vertices (see Figure 2.9).

A configuration is a connected subgraph of a completely triangulated planar graph that consists of an  $n$ -ring and the vertices and edges of the  $n$ -ring's interior (called the interior of the configuration) and such that the  $n$ -ring contains no diagonals. A subconfiguration is a connected subgraph of a configuration that is itself a configuration. The interior vertices of a configuration are said to be surrounded by the vertices of the  $n$ -ring. Configurations are classified by the ring size,

$n$ , and the number of interior vertices,  $m$ . Configurations are called  $n$ -ring configurations. Figure 2.3 shows a 9-ring configuration containing 3 interior vertices. Since the interior of a configuration is completely triangulated, the  $n$ -ring is uniquely determined by the degrees of the interior vertices and the edges that join these vertices. Thus, configurations are shown without the ring vertices using the following code (due to H. Heesch) for the degree of interior vertices:



Configurations do not contain vertices of degree less than 5 since they are reducible configurations (see examples of reduction procedures later). The configuration of Figure 2.3 is shown with and without its surrounding ring. A configuration is called reducible if it can be shown by certain specific methods to be described later that no graph containing this configuration can be minimal five chromatic. An algorithm which can be used to prove configurations reducible will be called a reduction procedure.

A set of graphs is called an unavoidable set if every graph must contain at least one element of this set as a subgraph. The following theorem defines one unavoidable set.

Theorem: In a connected, planar graph with no self loops or multiple edges, there is a vertex of degree  $\leq 5$ .

A concise proof of this theorem is found in Liu [16], p. 106. Let  $v_i$  denote a vertex of degree  $i$  and the  $i$  vertices it is adjacent to, then one unavoidable set is composed of 5 graphs:  $\{v_1, v_2, v_3, v_4, v_5\}$ .

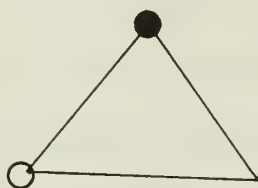
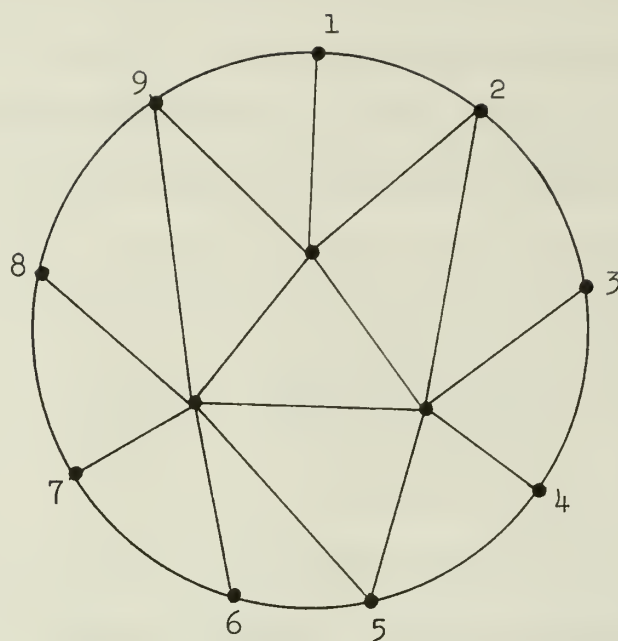
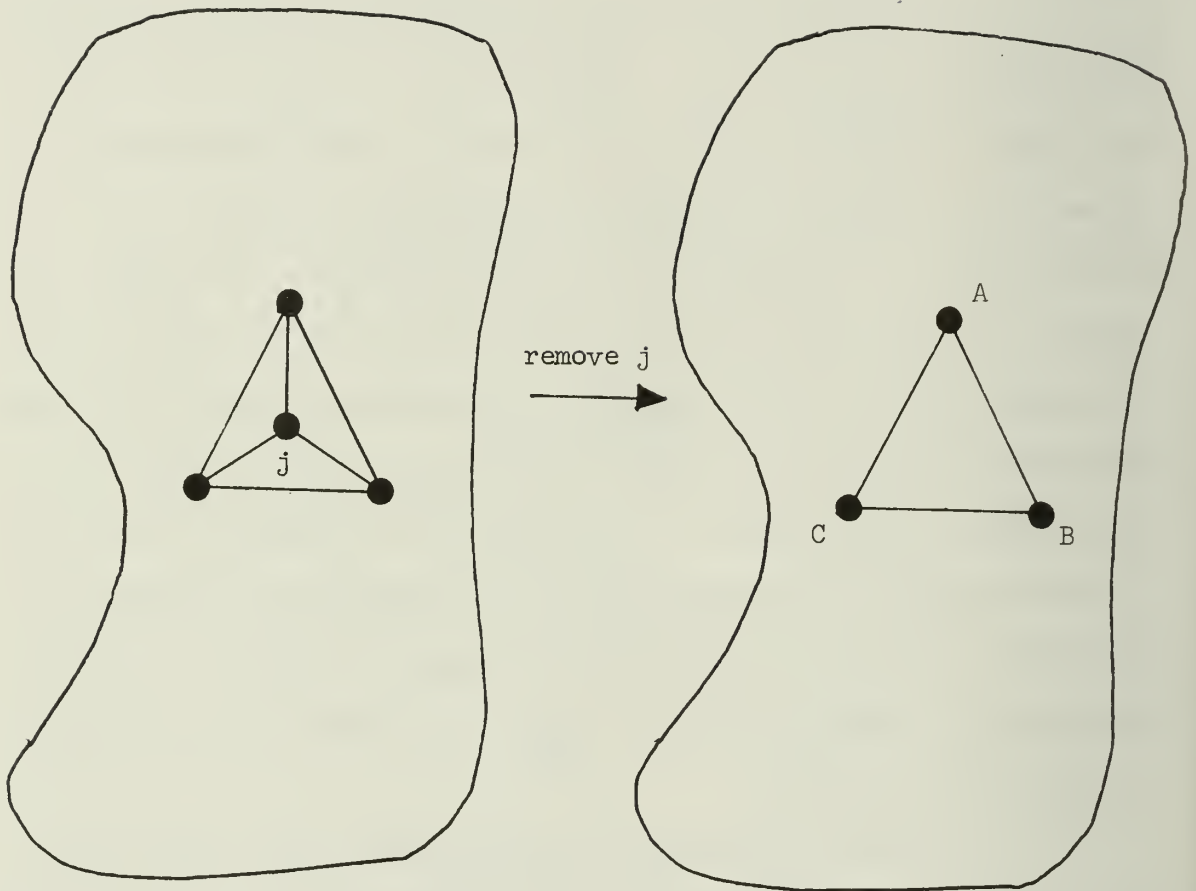


Figure 2.3 A 9-ring configuration with 3 interior vertices in two forms.

One possible way to prove the 4CC would be to show that all the elements of an unavoidable set are reducible configurations. Then a minimal five chromatic graph could not exist and the 4CC would be proved.

Two examples will be used to illustrate a reduction procedure that is known as D-reduction. First, consider the configuration that is a degree 3 vertex (surrounded by a 3-ring). This configuration is assumed to be a subgraph of a minimal five chromatic graph,  $Z$  (see Figure 2.4(a)).  $Z$  is assumed to consist of  $Q$  vertices. The degree 3 vertex,  $j$ , is removed and the remaining graph,  $Z'$ , of  $Q-1$  vertices is 4-colorable by definition. Suppose  $Z'$  is 4-colored. Since only 3 colors can appear on the 3-ring  $(A,B,C)$ ,  $j$  can always be replaced and colored the remaining (fourth) color. This produces a 4-coloring of  $Z$  which is a contradiction to the assumption that  $Z$  was minimal five chromatic. Therefore, a degree 3 vertex (and its surrounding 3-ring) is a reducible configuration. Single vertices of degree 1 and degree 2 can similarly be shown to be reducible.

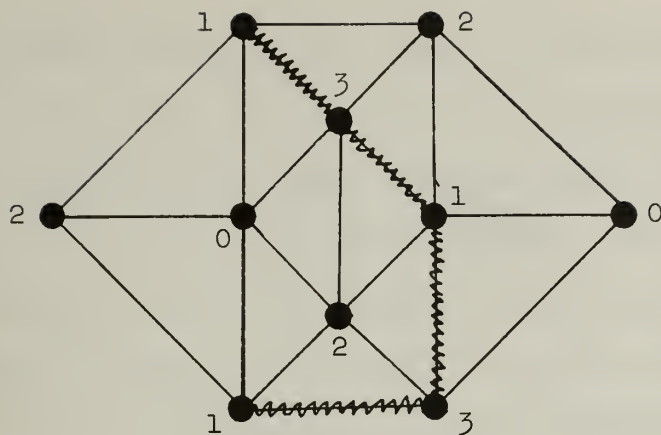
A few more definitions are required before the second example can be shown. The x-y skeleton of a four-colored graph,  $Z$ , is the graph consisting of all those vertices of  $Z$  that are colored either color  $x$  or color  $y$  and the edges that connect these vertices. The connected components of the x-y skeleton are called x-y components. For instance, there are 5 vertices and 4 edges in the 1-3 skeleton of Figure 2.5(a) and these form a single 1-3 component. There are two 0-2 components of  $Z$  (Figure 2.5(b)). An x-y interchange on an x-y component,  $I$ , of  $Z$  is the result of interchanging the colors  $x$  and  $y$  on the vertices of  $I$ . It is quite obvious that an x-y interchange applied to a proper coloring of  $Z$  produces a new proper coloring of  $Z$  (see Figure 2.5(c)).



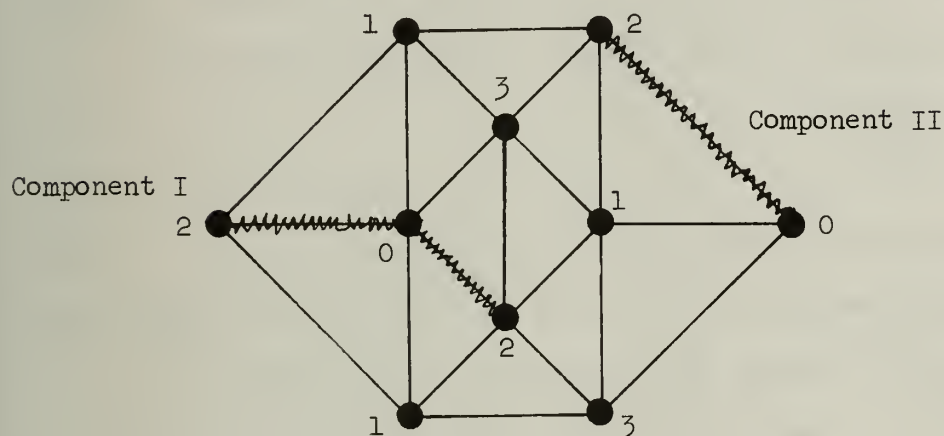
(a)  $Z$ , a minimal five chromatic graph of  $Q$  vertices that contains  $j$ , a degree 3 vertex

(b)  $Z'$ , the graph of  $Q-1$  vertices that remains when  $j$  is removed from  $Z$

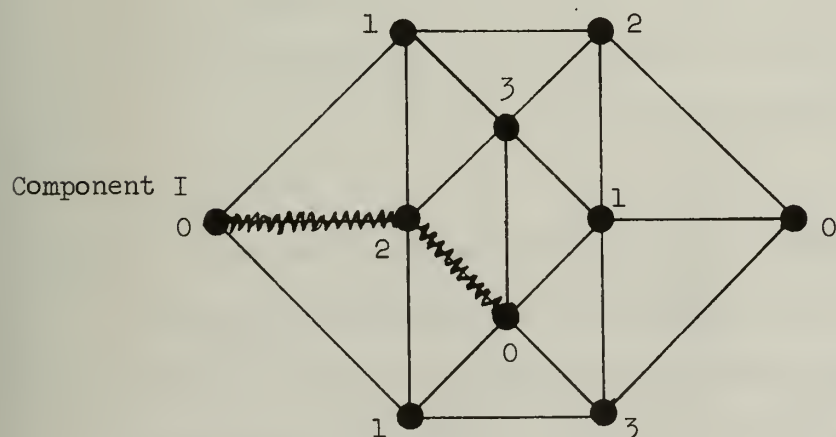
Figure 2.4 Reduction of a degree 3 vertex.



(a) The (unique) 1-3 component of a 4-colored graph,  $Z$ .



(b) Two 0-2 components, I and II, of  $Z$ .



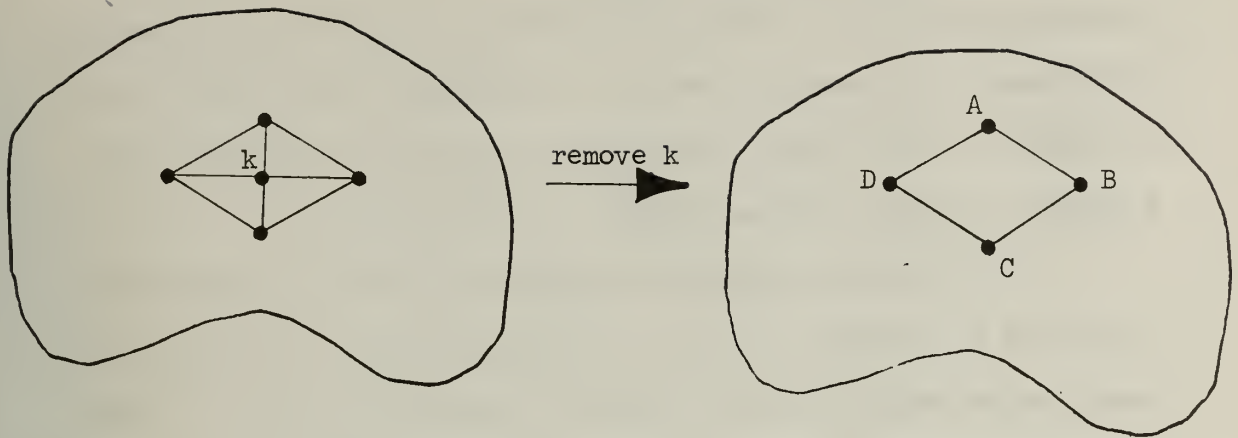
(c) A 0-2 interchange was performed on component I without destroying the proper 4-coloring or changing the colors of the vertices in component II.

Figure 2.5  $x$ - $y$  components and interchanges.

The second example will prove that a degree 4 vertex,  $k$ , is a reducible configuration.  $k$  is assumed to be a subgraph of a minimal five chromatic graph,  $Z$ .  $k$  is removed from  $Z$  producing a graph,  $Z'$ , of  $Q-1$  vertices that is then 4-colored (Figure 2.6(a) and (b)). All the colors that can appear on the 4-ring,  $A, B, C, D$ , are examined. If only 2 or 3 different colors are assigned to the vertices of the 4-ring,  $x$  can be replaced and colored with the remaining color. However, if all 4 colors appear on the 4-ring, then all the possible component structures of the graph outside the 4-ring must be considered. (Without loss of generality, the 4-ring,  $A, B, C, D$ , is colored as shown in Figure 2.6(c) and (d)).

Case 1: Vertices  $B$  and  $D$  are in the same 0-2 component of  $Z'$ . In other words, there is a path from  $B$  to  $D$  where the vertices are colored either 0 or 2. Since  $Z'$  is planar, there cannot be a path consisting of vertices from  $A$  to  $C$  that are colored 1 or 3. Thus,  $A$  and  $C$  are in different 1-3 components; and, a 1-3 interchange can be performed on the 1-3 component containing vertex  $C$  that will not change the color of vertex  $A$ . Since the color of vertex  $C$  has been changed to color 1, vertex  $x$  can be replaced and colored with color 3.

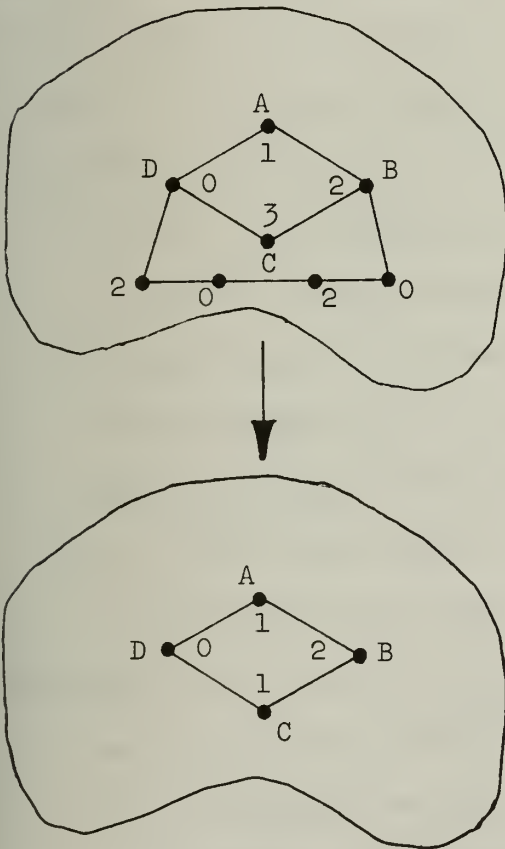
Case 2: Vertices  $B$  and  $D$  are in different 0-2 components of  $Z'$ . Thus, a 0-2 interchange can be performed on the 0-2 component containing vertex  $D$  without changing the color of vertex  $B$ . Vertex  $D$  having been changed to color 0, vertex  $x$  can then be replaced and colored color 2.



(a) A degree 4 vertex,  $k$ , in a minimal five chromatic graph,  $Z$ .

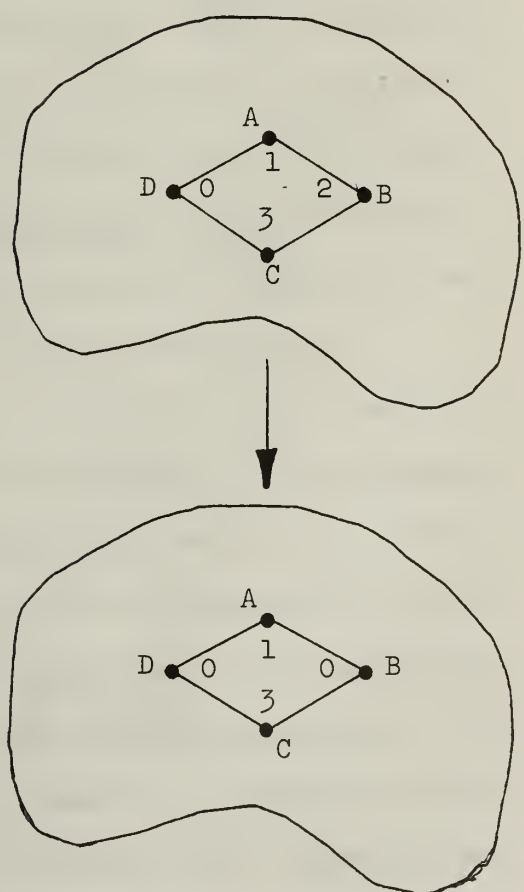
(b)  $Z'$ , a graph of  $Q-1$  vertices that is 4-colorable.

Case 1:



(c)  $B, D$  in same 0-2 component.

Case 2:



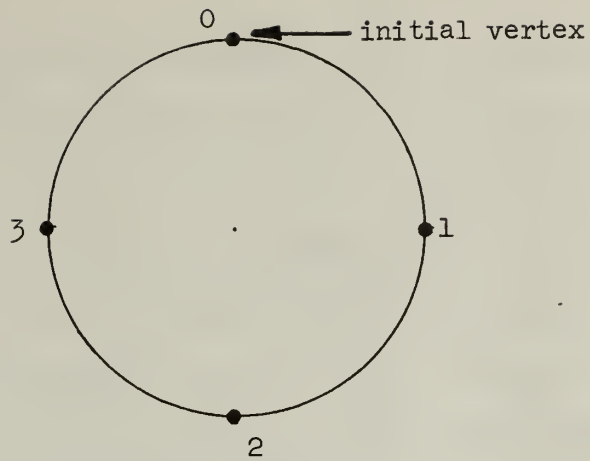
(d)  $B, D$  in different 0-2 component.

Figure 2.6 Reduction of a degree 4 vertex.

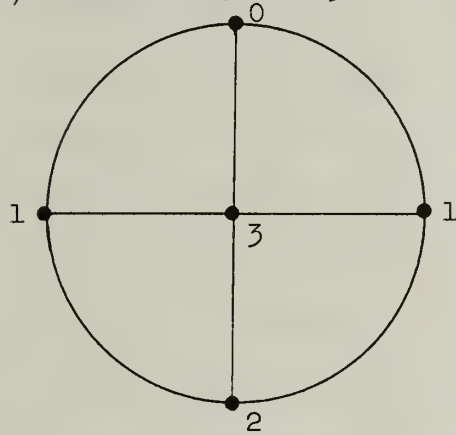
Therefore, when all 4 colors appear on the 4-ring, a color interchange can always be performed to produce a 3-coloring of the ring. With a 3-coloring of the 4-ring, vertex  $k$  can be replaced as before. Thus, a degree 4 vertex is also a reducible configuration.

The reduction procedure suggested by the last example was discovered by Kempe and is known as Kempe chaining. D-reduction is a reduction procedure (two algorithms are presented in Chapter 3) that uses Kempe chaining. In attempting D-reduction, all possible colorations of the  $n$ -ring that surrounds a configuration are considered. A (proper) coloration of an  $n$ -ring is determined by listing the colors of the vertices of the  $n$ -ring in clockwise order beginning with a specified initial vertex (see Figure 2.7(a)). The term "coloration" will be used to denote such a listing. A coloration is said to extend to the interior of an  $n$ -ring if there is at least one proper 4-coloring of the ring and its interior. A coloration extends to the exterior of an  $n$ -ring if there exists at least one proper 4-coloring of the ring and the exterior vertices.

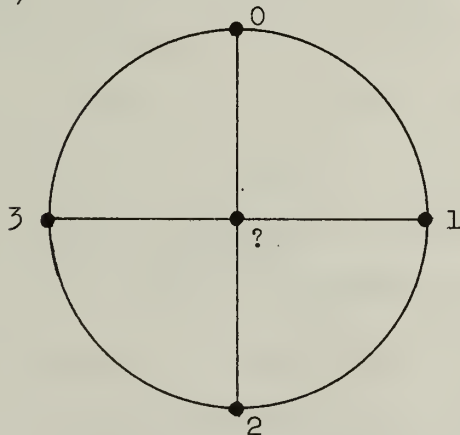
The set of all possible colorations of an  $n$ -ring (enumerated in Figure 7.3) is divided into several types of colorations when the D-reduction of a configuration,  $W$ , is attempted. A direct coloration of the unique  $n$ -ring surrounding  $W$  is a coloration that extends to the interior vertices of  $W$  (Figure 2.7(b)).  $D(W)$  is the set of direct colorations of  $W$ . An indirect coloration of the unique  $n$ -ring surrounding  $W$  is a coloration to which Kempe chaining can be applied to produce a direct coloration of  $W$  regardless of the component structure of the outside graph. The coloration '0123' in Figure 2.7(c) cannot lead to a proper 4-coloring of the interior, but '0123' can be changed by color interchanges into a 3-coloring of the 4-ring (see the reduction of a



(a) coloration: '0123'



(b) direct coloration: '0121'

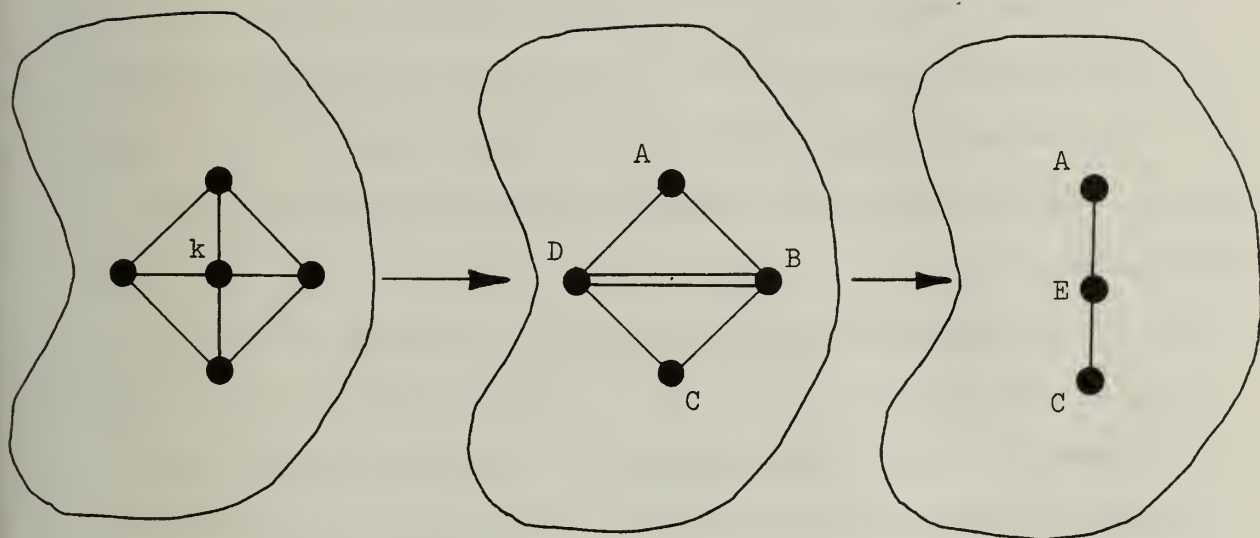


(c) indirect coloration: '0123'

Figure 2.7 Ring colorations.

degree 4 vertex above). Thus, colorations of the 4-ring that consist of 2 or 3 distinct colors are direct colorations while '0123' is an indirect coloration.  $I(W)$  is the set of indirect colorations of a configuration,  $W$ . The good colorations,  $G(W)$ , of  $W$  are all the colorations that are either direct or indirect colorations of  $W$ . Colorations that are not good colorations are called bad colorations. A configuration,  $W$ , is D-reducible if all possible colorations of its  $n$ -ring are in  $G(W)$ . D-reduction begins with  $D(W)$  and iteratively applies Kempe chaining to all colorations that are not yet good colorations until no more indirect colorations are found. A D-reducible configuration,  $W$ , can always be 4-colored no matter what coloration appears on the  $n$ -ring surrounding  $W$ .

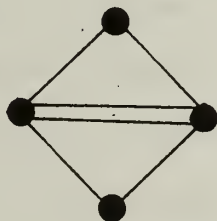
All possible colorations of an  $n$ -ring must be good colorations for a configuration to be D-reducible. However, it is possible to prove a configuration reducible by considering a subset of all the possible colorations of an  $n$ -ring. For example, there is another method that can be used to show that a degree 4 vertex is reducible. The degree 4 vertex,  $k$ , is again removed from a minimal five chromatic graph,  $Z$ , yielding  $Z'$  (see Figure 2.8(a) and (b)). This time vertices  $B$  and  $D$  are identified. When two vertices,  $B$  and  $D$ , are identified they are coalesced into a single vertex,  $E$  (see Figure 2.8(c)). The multiple edges that are produced when  $B$  and  $D$  are identified are replaced by single edges. (In this example, if a self-loop is produced at vertex  $E$ , then vertices  $A$  and  $C$  are coalesced.) Thus,  $B$  and  $D$  must have the same color in any 4-coloring of  $Z'$ . An identification of two vertices will be represented by a pair of parallel edges joining those two vertices (see Figure 2.8(b)). This identification may be thought of as a generalized edge that joins



(a) A degree 4 vertex,  $k$ , in  $Z$ , a minimal five chromatic graph

(b)  $Z'$

(c)  $Z''$



reducer

Figure 2.8 Reduction of a degree 4 vertex using a reducer.

two ring vertices. (No ambiguity results from using two parallel edges to represent an identification since multiple edges have been excluded.)

Now the direct colorations of the 4-ring can only contain 2 or 3 distinct colors.  $Z'$  can be 4-colored in the same way  $Z''$  is 4-colored using the color of vertex E as the color of vertices B and D. But a 2- or 3-coloring of the ring is a good coloration of W. Thus, k can always be replaced and colored the remaining color producing a 4-coloring of Z.

A generalized n-ring configuration is a graph consisting of an n-ring whose interior is completely triangulated using edges, vertices and generalized edges (identifications). There are no restrictions on diagonals in generalized configurations. A potential reducer of an n-ring configuration that has m interior vertices is defined to be a generalized n-ring configuration that contains fewer than m interior vertices. Therefore, whenever a configuration in a minimal five chromatic graph, Z, is replaced by a potential reducer, the resulting graph,  $Z'$ , has fewer vertices than Z, and  $Z'$  is 4-colorable. Since  $Z'$  is 4-colorable, some coloration of the potential reducer's n-ring must extend to the portion of  $Z'$  that is exterior to the n-ring. The only colorations that can extend to the portion of  $Z'$  that is exterior to the n-ring are the direct colorations of the potential reducer. Suppose every direct coloration of the potential reducer's n-ring is also a good coloration of the configuration's n-ring. Then, some good coloration of the configuration's n-ring must extend to the portion of Z that is exterior to the n-ring and, thus, the configuration is reducible. When the direct colorations of a potential reducer, P, are a subset of the good colorations of a configuration

W, then W is called C-reducible, and P is called an effective reducer of W. (Note that all potential reducers of a D-reducible configuration are effective reducers.) For the remainder of this thesis, potential reducers will be called reducers. When it is important to distinguish, the term effective reducer will be used. Whenever the direct colorations of a reducer, P, are not a subset of the good colorations of a configuration, W, then P is said to have failed to reduce W. Potential reducers can be constructed by placing identifications, diagonals and interior vertices inside an n-ring to produce a completely triangulated (in the sense of thinking of identifications as generalized edges) interior of the n-ring (see Figure 2.9).

A configuration, W, is irreducible when W is not D-reducible and when there are no effective reducers of W (all potential reducers fail). This thesis is concerned with testing reducibility of configurations by testing all of their potential reducers. A side result is that certain configurations are shown to be irreducible since they have no effective reducers. The generation of potential reducers is described in Chapters 4 and 5. Chapter 3 describes how the direct and indirect colorations of a configuration were obtained.

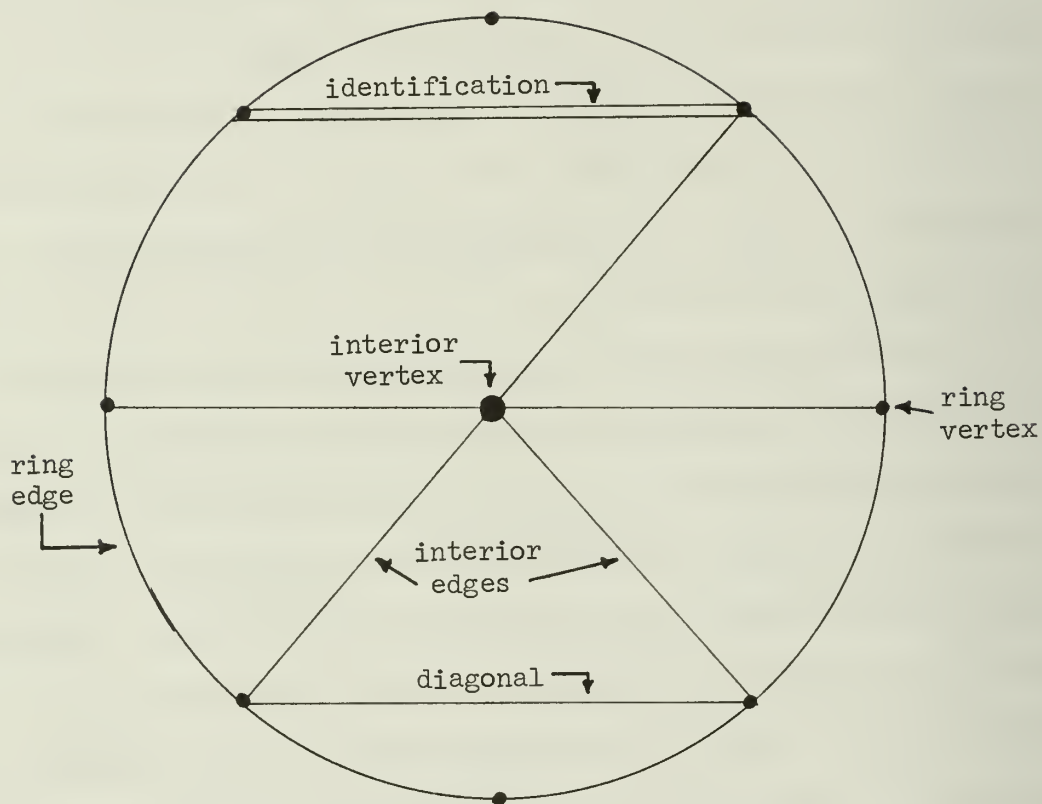


Figure 2.9 Potential reducer of an 8-ring configuration with  $\geq 2$  interior vertices.

## CHAPTER 3

## DIRECT AND INDIRECT COLORATIONS

In this chapter, the algorithms used to determine the direct and indirect ring colorations of a configuration are described. The canonical form and the data representation of a ring coloration are shown first.

The four "colors" used to color graph vertices are 0,1,2,3, because each of these numbers has a natural representation that uses only two binary digits. Given a ring of  $n$  vertices that was four-colored, the canonical form of this ring coloration was created as follows. The ring had a specified initial vertex and the vertices were numbered clockwise from #1 to # $n$ . Without loss of generality, the color of the initial vertex was called color 0. The color of vertex #2 (which had to be different from the color of vertex #1) was called color 1. Proceeding clockwise around the ring, the color of the next vertex not colored 0 or 1 was called color 2. The remaining color was called color 3. Multiple copies of the same ring coloration with different color names were thus eliminated from consideration.

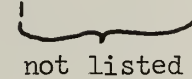
The following three consequences of putting every coloration into canonical form simplified the data representation of ring colorations. First, vertex # $n$  cannot be colored 0, since it is adjacent to vertex #1 which is always colored 0. Second, vertex #3 cannot be colored 1 or 3 for the following reasons. Vertex #3 is adjacent to vertex #2 which is

always colored 1; hence, vertex #3 cannot be colored 1. Whenever vertex #3 is not colored 0, its color cannot be called color 3 because of the definition of canonical form. Thus, vertex #3 must be colored either color 0 or color 2. Third, vertices #1 and #2 are always colored 0 and 1 respectively. Since the colors of vertices #1 and #2 remained constant, the colors of these vertices were not listed in the data representation of any ring coloration. Each of the remaining  $n-2$  vertices needed two bits to record its color. A single 32 bit word was used to store a ring coloration composed of 18 or fewer vertices.

It was desired that the data representation permit a rapid test of whether a coloration was "good". If a bit list area were formed in such a way that each ring coloration were the address of a bit, an area of  $4^{n-2}$  bits would have been needed. Since each computer byte contains  $2^3$  bits, this means  $2^{2n-7}$  bytes would have been required; for  $n=11$ ,  $2^{22-7} = 2^{15} = 32,768$  bytes.

Several techniques were used to significantly reduce the size of this bit list area. The address of the bit associated with a coloration consisted of a modification of the coloration itself. Since vertex #n could not have the same color as vertex #1, listing the coloration counter-clockwise from vertex #n meant that the addresses with vertex #n colored '00' could be ignored. In addition, the last bit in the representation of the allowable colors of vertex #3 ('00' and '10') was always a '0' and was ignored. Finally, only the addresses of bits between the smallest and largest possible colorations were required. Thus, when the number of vertices,  $n$ , of a ring coloration was even, the largest and smallest possible colorations were:

| vertex    | #n | #n-1 | #n-2 | #n-3 | ... | #6 | #5 | #4 | #3 | #2 | #1 |
|-----------|----|------|------|------|-----|----|----|----|----|----|----|
| largest:  | 11 | 10   | 11   | 10   | ... | 11 | 10 | 11 | 10 | 01 | 00 |
| smallest: | 01 | 00   | 01   | 00   | ... | 01 | 00 | 01 | 00 | 01 | 00 |


  
not listed

Considering the representations as hexadecimal numbers and subtracting:

$$\begin{array}{r}
 777\dots 7_{16} \\
 -222\dots 2_{16} \\
 \hline
 555\dots 5_{16} \\
 \underbrace{\hspace{1.5cm}} \\
 \frac{n-2}{2} \text{ digits}
 \end{array}$$

A similar analysis was performed when n was odd:

| vertex    | #n | #n-1 | #n-2 | #n-3 | ... | #5 | #4 | #3 |
|-----------|----|------|------|------|-----|----|----|----|
| largest:  | 11 | 10   | 11   | 10   | ... | 11 | 01 | 10 |
| smallest: | 01 | 00   | 01   | 00   | ... | 01 | 00 | 10 |

$$\begin{array}{r}
 1DD\dots DDB_{16} \\
 -088\dots 882_{16} \\
 \hline
 155\dots 559_{16} \\
 \underbrace{\hspace{1.5cm}} \\
 \frac{n-1}{2} \text{ digits}
 \end{array}$$

The area required for  $n=11$  now was  $15559_{16} = 87385 \text{ bits} = 10,922 \text{ bytes}$ , as opposed to 32,768 bytes. The regions for  $4 \leq n \leq 16$  are shown in Figure 3.1.

| <u>ring<br/>size</u> | <u>number of bits in<br/>bit list area</u> | <u>number of bytes in<br/>bit list area</u> |
|----------------------|--------------------------------------------|---------------------------------------------|
| 4                    | 5                                          | 1                                           |
| 5                    | 11                                         | 2                                           |
| 6                    | 85                                         | 11                                          |
| 7                    | 331                                        | 42                                          |
| 8                    | 1,365                                      | 171                                         |
| 9                    | 5,451                                      | 682                                         |
| 10                   | 21,845                                     | 2,731                                       |
| 11                   | 87,371                                     | 10,922                                      |
| 12                   | 349,525                                    | 43,691                                      |
| 13                   | 1,398,091                                  | 174,762                                     |
| 14                   | 5,592,405                                  | 699,051                                     |
| 15                   | 22,369,611                                 | 2,796,202                                   |
| 16                   | 83,886,080                                 | 10,485,761                                  |

Figure 3.1 Number of bytes required for  
bit list area.

The programs to find direct and indirect colorations both used the preceding ring coloration representation. There are other representations that would require a smaller bit area, but the time required to test a bit would have been increased. Since the chosen representation was a hashing technique, the bit associated with a coloration was found rapidly. Only 8 statements (IBM 360 Assembly language) were required to find and test a coloration's associated bit.

The program to obtain the direct colorations of a configuration, DIRECT, used a modified backtrack method. Each vertex in the configuration had four bits associated with it. These bits formed a "prohibition list" indicating which of the four colors could not be used to color the vertex. A section of macro-generated code was created for each vertex of the configuration. This code selected the color of the vertex and then went to the subroutine EXTEND. This subroutine extended the prohibition of the color selected to all the vertices adjacent to the vertex just colored. When a vertex had 3 colors prohibited, it was immediately colored with the remaining color. If a vertex had all 4 colors prohibited, the program, DIRECT, would backtrack since such a coloring of the configuration was impossible. After completely coloring the configuration, the ring coloration was recorded and the program backtracked to the last ring vertex colored. This prevented the generation of multiple colorings of the interior of a configuration for a single ring coloration.

Once the direct colorations were determined, the indirect colorations had to be generated by program DRED. D-reduction and the determination of indirect colorations were described briefly in Chapter 2.

The basic D-reduction algorithm, using multiple passes of the coloration list, is shown in Algorithm A1. By storing information in Steps H and K, it was possible to make only a single pass through the coloration list (Algorithm A2).

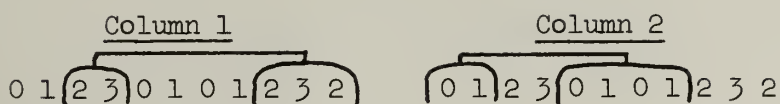
The implementation of Algorithm A1 (parts of which were also used to perform Algorithm A2) is described below. When a bad coloration was found, the component structure of the outside graph had to be considered. There are three ways to select color pairs from 4 colors: (01,23), (02,13), (03,12). Each color pairing determines the number of X-Y segments of a specific coloration. An X-Y segment is the longest string of vertices colored either X or Y. An X-Y segment is the portion of an X-Y component that is composed of vertices on the ring. For example,

Coloration: 01230101232

| <u>Color pairing</u> | <u>0-Y segments</u>   | <u>Number of<br/>0-Y segments</u> |
|----------------------|-----------------------|-----------------------------------|
| (01,23)              | 0 1 2 3 0 1 0 1 2 3 2 | 2                                 |
| (02,13)              | 0 1 2 3 0 1 0 1 2 3 2 | 5                                 |
| (03,12)              | 0 1 2 3 0 1 0 1 2 3 2 | 4                                 |

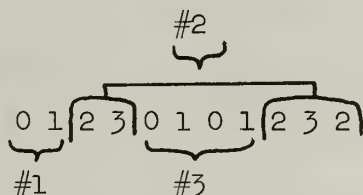
As in the D-reduction of a degree 4 vertex, every possible joining of these segments must be examined. One method of visualizing this process is to consider each color pairing to have an associated matrix of colorations. A column of a matrix corresponds to the choice of a particular segment structure. A segment structure is the manner that the X-Y segments of a coloration are joined. In the above example,

Color pairing (01,23)



The number of segments of one color pair determines the number of columns. Once a segment structure is chosen, the column is composed of color interchanges within segments (see Figure 3.2). If each column in the matrix of a color pairing contains a good coloration, then the original coloration is an indirect coloration.

All possible segment structures were encoded into segment tables in the implementation of Algorithms A1 and A2. In the example, two segments are joined, leaving 3 segments in which colors might be interchanged:



To insure that all possible interchanges were performed, a special binary reflected Gray code was used. By letting '1' indicate a color pair interchange from the original coloration, all possible combinations were obtained using the following portion of the Gray code:

| <u>position<br/>interchanged</u> | <u>Column 1</u>                                    | <u>Color pair interchanges<br/>performed to restore<br/>coloration to canonical form</u> |
|----------------------------------|----------------------------------------------------|------------------------------------------------------------------------------------------|
| --                               | 0 1 (2 3) 0 1 0 1 (2 3 2)                          |                                                                                          |
| 1                                | <u>1</u> 0 (2 3) 0 1 0 1 (2 3 2)                   | $0,1 \rightarrow$ 0 1 2 3 1 0 1 0 2 3 2                                                  |
| 2                                | 1 0 ( <u>3 2</u> ) 0 1 0 1 ( <u>3 2 3</u> )        | $0,1;2,3 \rightarrow$ 0 1 2 3 1 0 1 0 2 3 2                                              |
| 1                                | <u>0</u> 1 ( <u>3 2</u> ) 0 1 0 1 ( <u>3 2 3</u> ) | $2,3 \rightarrow$ 0 1 2 3 0 1 0 1 2 3 2                                                  |
| 3                                | 0 1 ( <u>3 2</u> ) <u>1 0 1 0</u> ( <u>3 2 3</u> ) | $2,3 \rightarrow$ 0 1 2 3 1 0 1 0 2 3 2                                                  |
| 1                                | <u>1</u> 0 ( <u>3 2</u> ) 1 0 1 0 ( <u>3 2 3</u> ) | $0,1;2,3 \rightarrow$ 0 1 2 3 0 1 0 1 2 3 2                                              |
| 2                                | 1 0 ( <u>2 3</u> ) 1 0 1 0 ( <u>2 3 2</u> )        | $0,1 \rightarrow$ 0 1 2 3 0 1 0 1 2 3 2                                                  |
| 1                                | <u>0</u> 1 ( <u>2 3</u> ) 1 0 1 0 ( <u>2 3 2</u> ) |                                                                                          |

Figure 3.2 Colorations within a column.

| position | <u>1</u> | <u>2</u> | <u>3</u> | <u>position changed</u> |
|----------|----------|----------|----------|-------------------------|
|          | 0        | 0        | 0        | original                |
|          | 1        | 0        | 0        | 1                       |
|          | 1        | 1        | 0        | 2                       |
|          | 0        | 1        | 0        | 1                       |
|          | 0        | 1        | 1        | 3                       |
|          | 1        | 1        | 1        | 1                       |
|          | 1        | 0        | 1        | 2                       |
|          | 0        | 0        | 1        | 1                       |

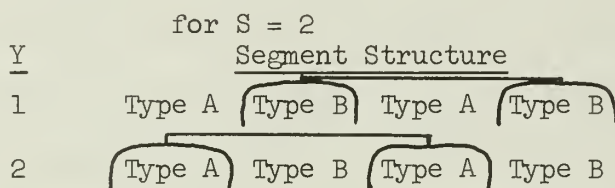
This code also had the advantage that between each number, only one position was changed. Thus, a minimum amount of work had to be done to generate the colorations in a column. In addition, the requirement that all ring colorations be kept in canonical form saved some work. For instance, there are really only two different colorations in Figure 3.1, 0 1 2 3 0 1 0 1 2 3 2 (original) and 0 1 2 3 1 0 1 0 2 3 2.

Once a coloration in a column was generated, it was checked to see if it was good. If the coloration was bad, the generation of the columns continued. If the coloration was good, the column was marked as a good column and work proceeded with the next column. If all of the columns of a certain color pairing were good, the original coloration was marked as a good coloration. A coloration remained bad if there was at least one bad column in each of the three color pairing rows.

Successive iterations of Algorithm A1 were performed until an iteration was reached in which no colorations were made good. Since further iterations could not then make any new colorations good, the procedure terminated. Only a single iteration of Algorithm A2 was needed, since information was stored in the data structure described below.

LIST is a list of all possible colorations of an n-ring. When Algorithm A1 begins, LIST also indicates whether a coloration is a direct coloration of the configuration. A portion of each of the tables used in this algorithm is shown below:

| Gray code<br>position indicator |             | Number of<br>Segments | Number of<br>Columns |
|---------------------------------|-------------|-----------------------|----------------------|
| <u>k</u>                        | <u>G(k)</u> | <u>S</u>              | <u>COL(S)</u>        |
| 1                               | 1           | 1                     | 2                    |
| 2                               | 2           | 2                     | 5                    |
| 3                               | 1           | 3                     | 14                   |
| 4                               | 3           | 4                     | 42                   |
| 5                               | 1           | 5                     | 132                  |
| 6                               | 2           |                       |                      |
| 7                               | 1           |                       |                      |



G(k) is the number of a segment in a coloration, W. The actual vertex positions of W that are to participate in a color pair interchange are determined by the segmentation of W and the way these segments are joined (indicated by Y).

Step A Obtain the next bad coloration, C, from LIST.

Step B Set S ← number of segments created in C by the color pairing (01,23), i ← 1. Go to Step E.

Step C Set S ← number of segments created in C by the color pairing (02,13), i ← 2. Go to Step E.

Step D Set S ← number of segments created in C by the color pairing (03,12), i ← 3. Go to Step E.

Step E If S = 1, go to Step I. Set Y ← 0, COUNT ← 0 and MAX ← COL(S) (maximum number of columns).

Step F If Y > MAX, go to Step J. Y indicates the segment structure of C to be used for a column. Set Y ← Y+1, W ← C, k ← 1.

Step G (Color interchange)  
Determine the positions to be interchanged in W from G(k), Y and the segmentation of C. Perform the color interchange on W producing a coloration X.

Step H If  $X$  is a good coloration, then set  $COUNT \leftarrow COUNT+1$  and go to Step F. Otherwise, set  $k \leftarrow k+1$ . If  $k > 2^{S+1}-1$ , go to Step I. Set  $W \leftarrow X$  and go to Step G.

Step I (All colorations in a column are bad)  
If  $i = 1$ , go to Step C.  
If  $i = 2$ , go to Step D.  
If  $i = 3$ , go to Step K.

Step J If  $COUNT = MAX$ , mark  $C$  as a good coloration.

Step K If there are more bad colorations below  $C$  in  $LIST$ , go to Step A. If  $GOOD = 1$ , set  $GOOD \leftarrow 0$  and start another pass through  $LIST$ , go to Step A. Otherwise, stop.

Algorithm A1 Multiple pass D-reduction.

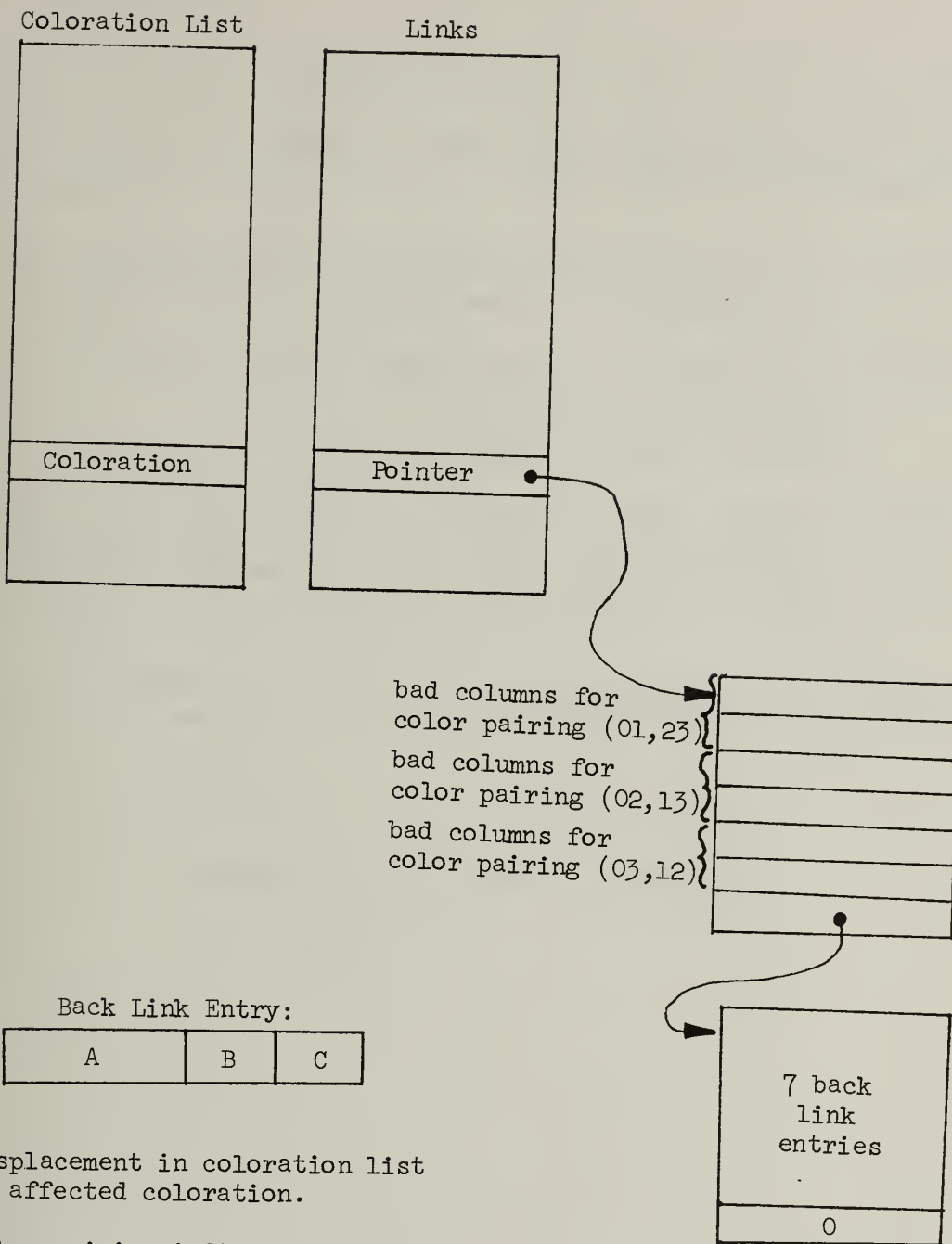
The data structure shown in Figure 3.3 was built up for every coloration that was not made good by the D-reduction algorithm. A list of columns that remained bad was kept. For every coloration in a bad column, a back link entry was created which pointed to the bad column. If a coloration became good, its back link entries were followed. In some cases, the new good coloration decreased the number of bad columns of a previous coloration. If all the columns of a color pairing of a coloration became good, the coloration itself became good.

Both Algorithms A1 and A2 had advantages. Algorithm A2 generally performed faster than Algorithm A1 in the following cases:

1. A configuration with ring size less than 10.
2. For  $n = 10, 11$  when the number of bad colorations was less than 50%.
3. For  $n = 10, 11$  any configuration that was D-reducible.

For those configurations with ring size greater than 9 that had a large number of bad colorations, the size of the area required to store Algorithm A2's back links became prohibitive (for  $n = 9$ , 242,000 bytes of core storage was normally required; for  $n = 10$ , approximately three times the storage of  $n = 9$  would have been required). The implementation of Algorithm A1 using the segment tables and the ring coloration representation proved to be rapid enough for use through rings of size 14 (see Chapter 8). Algorithm A1 was thus the choice for rings of size greater than 9. Algorithm A2 was used when the ring size was less than 10.

If some bad colorations remained after the direct and indirect colorations were determined, C-reduction was attempted. A complete set



A: Displacement in coloration list to affected coloration.

B: Color pairing indicator:

'00' -- (01,23)

'08' -- (02,13)

'10' -- (03,12)

C: Affected column number (1-42).

Figure 3.3 Data structure used by Algorithm A2.

The following steps are changed from Algorithm A1 to produce Algorithm A2.

- Step A If the end of the coloration list has been reached, stop.  
Otherwise, obtain the next bad coloration, C, from the list.
- Step H If X is good, set  $COUNT \leftarrow COUNT+1$  and go to Step F.  
Otherwise put (X,Y,i) on STACK,  $k \leftarrow k+1$ .  
If  $k > 2^S-1$ , go to Step L.  
Otherwise, set  $W \leftarrow X$  and go to Step G.
- Step J If  $COUNT \neq MAX$ , go to Step I. Otherwise, mark C as a good coloration. Follow all C's back link entries making the appropriate columns good. If all the columns of a color pairing of a coloration, Z, became good, set  $C \leftarrow Z$  and repeat Step J.
- Step K For each triple (X,Y,i) on STACK, create a back link entry for C. Go to Step A.

Algorithm A2 Single pass D-reduction.

of reducers to use in C-reduction was needed in order to show irreducibility. The creation of these reducers is described in the next two chapters.

## CHAPTER 4

## REDUCER GENERATION

, All potential reducers of configurations with ring size  $6 \leq n \leq 11$  and with  $16-n$  interior vertices were generated. These potential reducers form the complete set of reducers,  $R(n,m)$ , with  $6 \leq n \leq 11$  and  $m \leq 15 - n$ . The algorithm to generate all potential reducers was designed to eliminate as many redundant reducers as possible. A reducer, B, is called redundant when the set of direct colorations of B is a superset of the set of direct colorations of another reducer, A. In other words, B is redundant since A can replace B in any reducibility or irreducibility proofs.

Lemma 4.1: A redundant reducer can be eliminated from the complete set of reducers  $R(n,m)$ .

Proof: Let X,Y be reducers with  $D(X) \subseteq D(Y)$ .

Case 1: Let W be a configuration that is reduced by Y. Then  $D(Y) \subseteq G(W)$ . But,  $D(X) \subseteq D(Y) \subseteq G(W)$ . Therefore, whenever Y reduces a configuration, X also reduces it.

Case 2: Assume X does not reduce W. Thus, there is at least one coloration  $c$ ,  $c \in D(X)$ , such that  $c \notin G(W)$ . Since  $D(X) \subseteq D(Y)$ ,  $c \in D(Y)$ . Therefore, if X does not reduce W, Y cannot reduce W.

Since  $X$  has a smaller direct set, it is clearly more efficient to use  $X$  in place of  $Y$  in all reducibility computations. Thus,  $Y$  can be eliminated from  $R(n,m)$ .

The following restrictions prevented a number of redundant reducers from being generated.

Lemma 4.2: A reducer with an interior which is not completely triangulated is redundant.

Proof: Let  $R$  be a reducer whose interior is not completely triangulated. Let  $T$  be a completely triangulated reducer obtained by adding all possible interior edges to  $R$ . Let  $T$  be four-colored; then, transform  $T$  to  $R$  by removing the appropriate edges. Removing an edge lessens the restrictions on the allowable colors for the two vertices on that edge. Thus, every four coloring of  $T$  is a four coloring of  $R$  and  $D(T) \subseteq D(R)$ . Therefore,  $R$  is redundant.

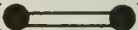
Lemma 4.3: A reducer with a reducible sub-configuration is redundant.

Proof: Let  $R$  be a reducer containing  $X$ , a sub-configuration which is reduced by  $Y$ . Thus,  $D(Y) \subseteq G(X)$ . Let  $S$  be the reducer obtained by replacing  $X$  by  $Y$  in  $R$ .



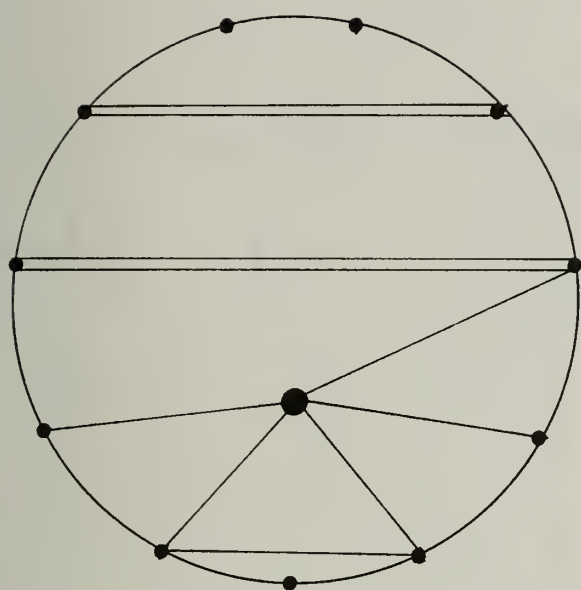
Since  $D(Y) \subseteq G(X)$ , all four colorings of  $R$  are also four colorings of  $S$ .  $D(S) \subseteq D(R)$ . Therefore,  $R$  is redundant.

Corollary 4.1: A reducer with a vertex of degree less than 5 is redundant (see Chapter 2).

For these reasons, all the reducers that were generated were completely triangulated, and all interior vertices had degree 5 or more. To each reducer a graph was associated. To obtain the graph corresponding to a reducer, all pairs of identified vertices () were collapsed into single vertices (see Figure 4.1). Collapsing vertices produced articulation vertices in the corresponding graphs. An articulation vertex is a vertex whose removal causes the graph to become disconnected. The articulation vertices of a reducer will be called collapsed vertices. (Collapsed vertices in the graph represent 2 or more ring vertices in the corresponding reducer.)

The graphs of reducers in  $R(n,m)$  are elements of the set of all connected planar graphs with at most  $n+m$  vertices. Since any planar graph can be generated by starting with one of its spanning trees and adding edges, the reducers could have been generated by adding edges to spanning trees. A spanning tree of a graph,  $G$ , is a connected subgraph of  $G$  that has no circuits but contains all the vertices of  $G$ .

This method of creating reducers by adding edges to spanning trees will be called the basic algorithm for generating reducers. The basic algorithm is displayed in Figure 4.2. The set of all spanning trees of  $v$  vertices is the top entry in column  $v$  of Figure 4.2. Each element of the second set in a column was obtained by adding a single edge to an element of the top set at some position which would create a triangulated reducer. Because the reducers must be completely triangulated, an edge could only be added between two ring vertices that were separated by exactly one other vertex on the reducer's ring. Adding an edge caused the reducer's ring size,  $n$ , to decrease by 1. The table of reducer ring sizes determined by the number of vertices,  $v$ , and the number of edges,  $e$ , in the graphs is shown in Figure 4.3.

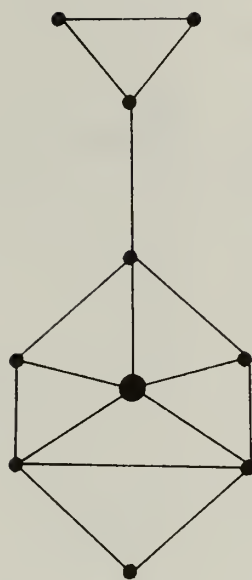


Reducer

$$n = 11$$

$$m = 1$$

$$v = 12$$



Graph

Figure 4.1 Reducer and its corresponding graph.

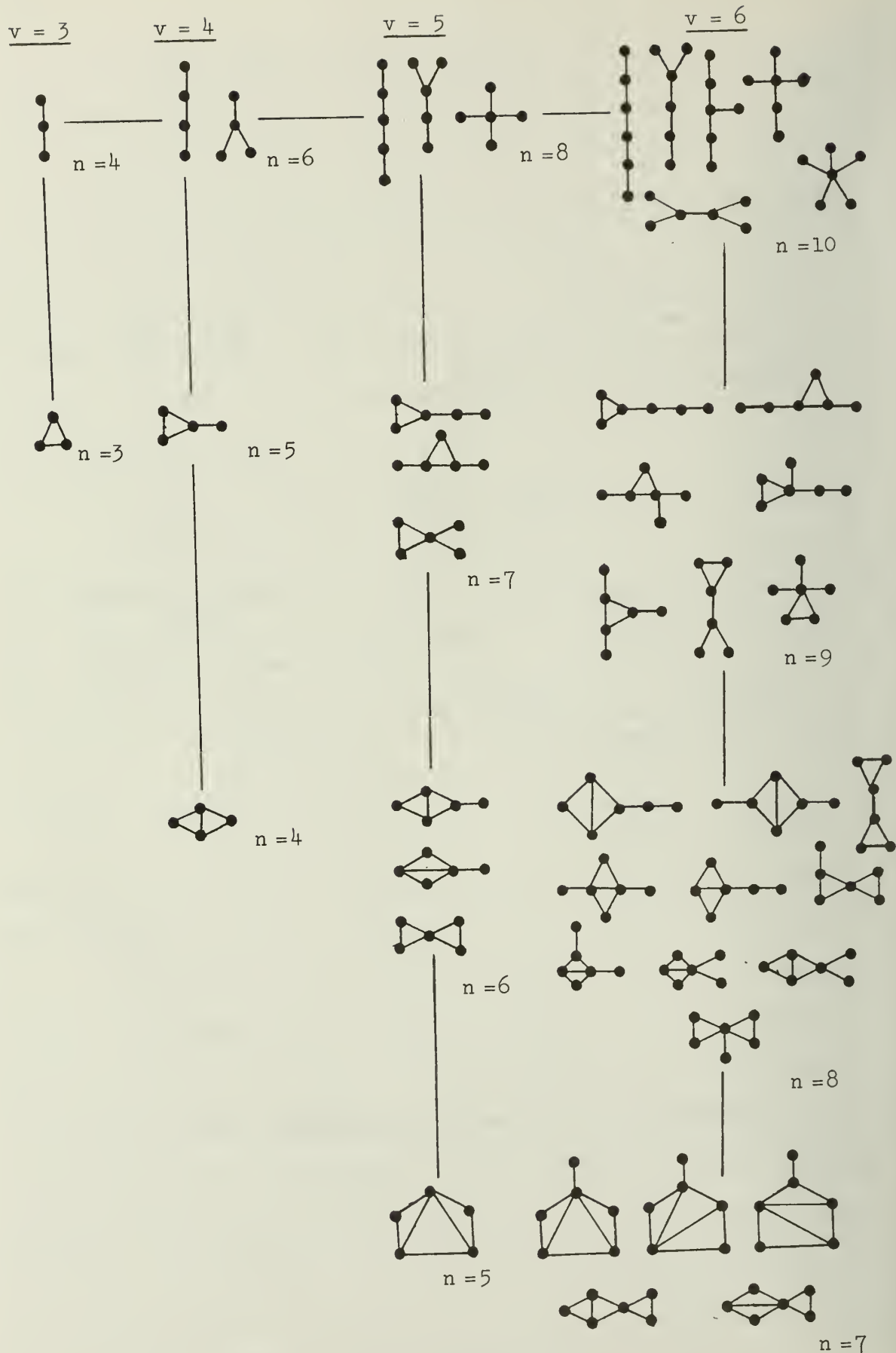


Figure 4.2 Generation of graphs from spanning trees.

| $e \backslash v =$ | <u>4</u> | <u>5</u> | <u>6</u> | <u>7</u> | <u>8</u> | <u>9</u> | <u>10</u> | <u>11</u> | <u>12</u> | <u>13</u> | <u>14</u> | <u>15</u> |
|--------------------|----------|----------|----------|----------|----------|----------|-----------|-----------|-----------|-----------|-----------|-----------|
| $v-1$              | 6        | 8        | 10       | 12       | 14       | 16       | 18        | 20        | 22        | 24        | 26        | 28        |
| $v$                | 5        | 7        | 9        | 11       | 13       | 15       | 17        | 19        | 21        | 23        | 25        | 27        |
| $v+1$              | 4        | 6*       | 8        | 10       | 12       | 14       | 16        | 18        | 20        | 22        | 24        | 26        |
| $v+2$              |          | 5        | 7        | 9        | 11       | 13       | 15        | 17        | 19        | 21        | 23        | 25        |
| $\vdots$           |          |          | 6*       | 8        | 10       | 12       | 14        | 16        | 18        | 20        | 22        | 24        |
|                    |          | 5        | 7        | 9        | 11       | 13       | 15        | 17        | 19        | 21        | 23        |           |
|                    |          |          | 6*       | 8        | 10       | 12       | 14        | 16        | 18        | 20        | 22        |           |
|                    |          |          |          | 7        | 9        | 11       | 13        | 15        | 17        | 19        | 21        |           |
|                    |          |          |          | 6*       | 8        | 10       | 12        | 14        | 16        | 18        | 20        |           |
|                    |          |          |          |          | 7        | 9        | 11        | 13        | 15        | 17        | 19        |           |
|                    |          |          |          |          | 6        | 8        | 10        | 12        | 14        | 16        | 18        |           |
|                    |          |          |          |          |          | 7        | 9         | 11        | 13        | 15        | 17        |           |
|                    |          |          |          |          |          | 6        | 8         | 10        | 12        | 14        | 16        |           |
|                    |          |          |          |          |          |          | 7         | 9         | 11        | 13        | 15        |           |
|                    |          |          |          |          |          |          | 6         | 8         | 10        | 12        | 14        |           |
|                    |          |          |          |          |          |          | 5         | 7         | 9         | 11        | 13        |           |
|                    |          |          |          |          |          |          |           | 6         | 8         | 10        | 12        |           |
|                    |          |          |          |          |          |          |           |           | 7         | 9         | 11        |           |
|                    |          |          |          |          |          |          |           |           |           | 8         | 10        |           |
|                    |          |          |          |          |          |          |           |           |           |           |           | 9         |

Figure 4.3 Table of ring sizes for reducers given the number of vertices,  $v$ , and the number of edges,  $e$ .

The following definitions are needed for the subsequent discussion and lemmas.  $S(v,n)$  is the set of all reducers of ring size  $n$  whose graphs contain exactly  $v$  vertices. Each entry in Figure 4.3 represents one set  $S(v,n)$ .  $S'(v,n)$  is the subset of  $S(v,n)$  consisting of reducers with no interior vertices. Finally,  $S_i(v,n)$  is the subset of  $S(v,n)$  in which each reducer has an uncollapsed ring vertex of degree less than or equal to  $i$ . The degree of a ring vertex,  $x$ , is determined by the number of interior edges and ring edges that are incident with  $x$ .

The set of potential reducers of a configuration with ring size  $n$  and with  $m$  interior vertices is the union of the sets  $S(v,n)$  where  $v-n \leq m$ . For example, the reducers of a configuration with  $n=6$  and  $m=3$  is the union of sets  $S(4,6)$ ,  $S(5,6)$ ,  $S(6,6)$ ,  $S(7,6)$ ,  $S(8,6)$ . These sets are starred (\*) in Figure 4.3.

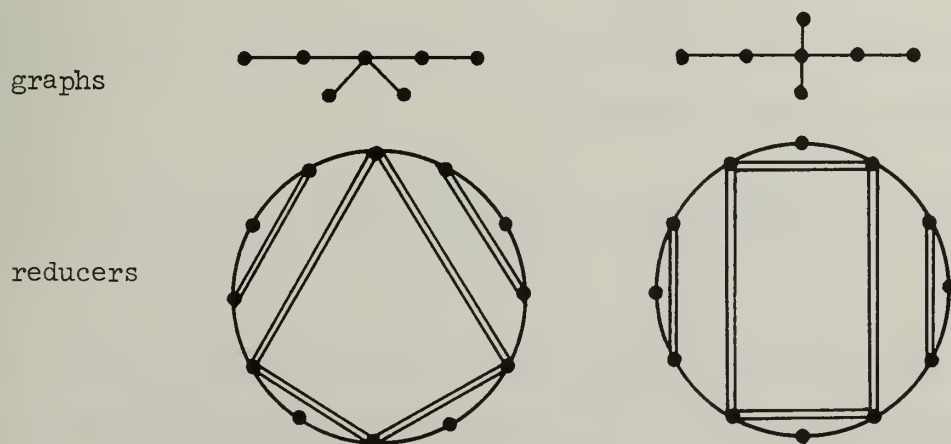
A decision was made to generate potential reducers only for configurations with  $n \leq 11$ . Colorations of 11-rings could have a maximum of 4 segments in D-reduction. Increasing the ring size to 12 would have required the D-reduction program to handle up to 5 segments. In addition, there are  $C_{11} = 7,381$  possible colorations for  $n=11$  while  $C_{12} = 22,144$ . Less than half the disk storage space needed for a bit string (see Chapter 6) of good colorations for a reducer with  $n=12$  was required for a reducer with  $n=11$  (2,768 versus 923 bytes). Similarly, the hashing technique of Chapter 3 required 10,922 bytes for  $n=11$  and 43,691 bytes when  $n=12$ . Finally, a listing of the colorations of an 11-ring using 4 bytes per coloration was written to disk in one contiguous block (29,424 bytes); however, such a listing for 12-ring exceeded the maximum number of bytes (32,768 bytes) that could be written in a single record. For these reasons, the ring size was limited to  $n \leq 11$ . The number of

vertices in a reducer was restricted to  $v \leq 15$  by limitations of time and core storage (discussed in Chapter 5).

Since the ring size was limited to  $n \leq 11$ , the desired reducers are in the sets below and to the left of the double line in Figure 4.3. If the basic algorithm had been used, all of the sets in Figure 4.3 would have had to be generated and stored. Several estimates were made of the number of graphs in the sets  $S(v, n)$  where  $v \leq 15$  but  $n > 11$ . One estimate was to lower bound the number of spanning trees of a graph with  $v$  vertices (the sets in the top row of Figures 4.2 and 4.3). This lower bound was obtained by examining the count of unrooted, unlabeled trees of  $v$  vertices:

| $v$                | 4 | 5 | 6 | 7  | 8  | 9  | 10  | 11  | 12  | 13   | 14   | 15   |
|--------------------|---|---|---|----|----|----|-----|-----|-----|------|------|------|
| number<br>of trees | 2 | 3 | 6 | 11 | 23 | 47 | 106 | 235 | 551 | 1301 | 3159 | 7741 |

This count was only a lower bound on the number of elements in  $S(v, 2v-2)$  for the following reason. When  $v = 7$ , one of the eleven unrooted, unlabeled trees actually represented the following two spanning trees (and their corresponding reducers):



It is not known if this lower bound is close to the actual number of spanning trees since the sets  $S(v, 2v-2)$  for  $v > 7$  were not generated. The second estimate was the number of elements in  $S'(n, n)$  (calculated by F. Bernhart [6] and shown in Figure 4.4). Since  $S'(15, 15)$  is a subset of  $S(15, 15)$ , there are at least  $2^{4,833}$  elements in  $S(15, 15)$ . Thus, the time and disk storage space required to generate all the sets represented in Figure 4.3 would have been prohibitive.

Several modifications were made to the basic algorithm which permitted only the sets  $S(v, n)$  where  $v \leq 15$  and  $n \leq 11$  to be generated (these sets are represented by entries below and to the left of the double line in Figure 4.3). The following seven lemmas are used to show that all the stated reducers were generated. The lemmas are followed by the modifications of the basic algorithm (Steps 1-4).

Lemma 4.4: A reducer with no collapsed or interior vertices has a ring vertex of degree 2.

Proof: Since the interior of a reducer in  $S'(n, n)$  is completely triangulated, there are  $n-2$  triangular faces by Euler's formula. Each triangular face is bordered by 0, 1, or 2 of the  $n$  ring edges. By the pigeonhole principle, there must be at least one face that is bordered by 2 ring edges. The vertex adjacent to those two ring edges is the desired ring vertex of degree 2.

Lemma 4.5: A reducer in  $S(v, i)$ ,  $2v-6 \leq i \leq 2v-2$ , contains no interior vertex.

Proof: Let  $x$  be the graph of a reducer in  $S(v, i)$ ,  $2v-6 \leq i \leq 2v-2$ . Thus,  $x$  has fewer than  $v+4$  edges. In other words, fewer than 5 edges are added

|       |   |   |   |    |    |     |     |      |      |        |        |
|-------|---|---|---|----|----|-----|-----|------|------|--------|--------|
| $n$   | 3 | 4 | 5 | 6  | 7  | 8   | 9   | 10   | 11   | 12     | 13     |
| $T_n$ | 1 | 2 | 5 | 14 | 42 | 132 | 429 | 1430 | 4862 | 16,796 | 58,786 |
| $F_n$ | 1 | 1 | 1 | 3  | 4  | 12  | 27  | 82   | 228  | 733    | 2,282  |

|       |         |         |           |           |
|-------|---------|---------|-----------|-----------|
| $n$   | 14      | 15      | 16        | 17        |
| $T_n$ | 208,012 | 742,900 | 2,674,440 | 9,694,845 |
| $F_n$ | 7,528   | 24,833  | 83,897    | 285,357   |

Figure 4.4  $F_n$  = the number of elements in  $S'(n,n)$ .  
 $T_n$  = the number of elements in  $S'(n,n)$   
without isomorph rejection.

to a spanning tree of  $x$ . Since each interior vertex must have degree greater than 4 (Corollary 4.1), at least five edges must be added to a spanning tree to form an interior vertex. Therefore,  $x$  has no interior vertex. No reducer in the sets above the dashed line in Figure 4.3 has an interior vertex.

Removing a non-disconnecting ring edge from a graph in  $S(v, n)$  results in a graph in  $S(v, n+1)$ .



The consequence of removing such an edge was to move to the next higher set in a column of Figure 4.3.

Lemma 4.6:  $S(v, n+1)$  can be obtained from  $S(v, n)$ .

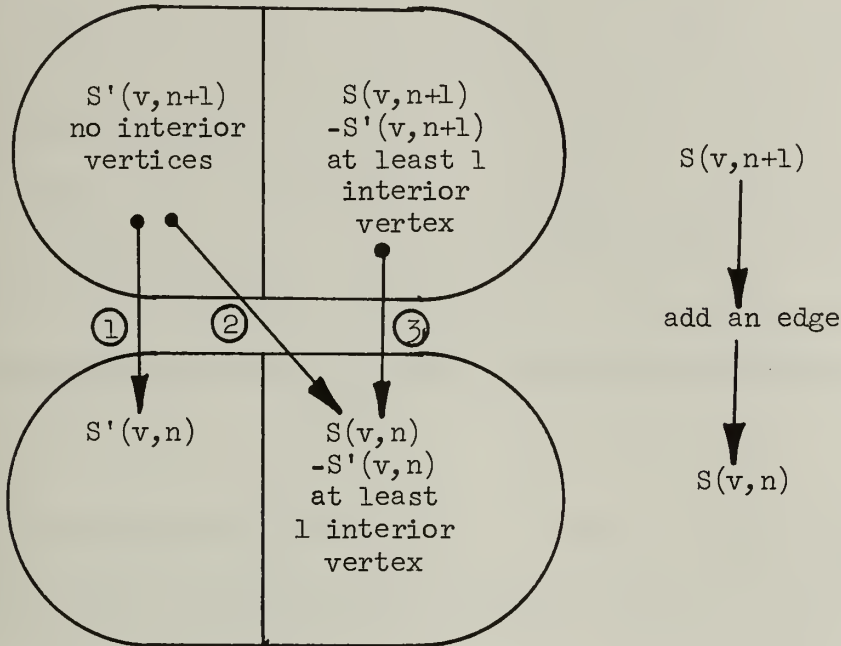
Proof: When generating graphs from their spanning trees in the basic algorithm, a graph in  $S(v, n)$  is generated by adding a single edge to a graph in  $S(v, n+1)$ . This process can be reversed to generate  $S(v, n+1)$  given  $S(v, n)$ . A non-disconnecting ring edge is removed from a graph in  $S(v, n)$  to obtain a graph in  $S(v, n+1)$ .

Lemma 4.7: Each graph in  $S'(v, n)$ ,  $v < n$ , contains a collapsed vertex.

Proof: Since the number of vertices in the graph is less than the reducer's ring size, there must be at least one collapsed vertex in the graph.

Lemma 4.8: If  $v \leq n$ ,  $S'(v, n+1)$  can be obtained from  $S'(v, n)$ .

Proof: The process of generating graphs from their spanning trees in the basic algorithm at the stage when  $S(v, n+1)$  goes to  $S(v, n)$  is examined.



Graphs in  $S(v, n+1)$  with interior vertices can only generate graphs in  $S(v, n) - S'(v, n)$ ; therefore, arrow ③ can be ignored. Arrow ① can be reversed. The one concern is that there might exist a graph,  $x \in S'(v, n+1)$ , that only generates elements in  $S(v, n) - S'(v, n)$  (arrow ②). However, by Lemma 4.7 and because  $v$  is less than  $n+1$ , every graph in  $S'(v, n+1)$  has a collapsed vertex.

Adding an edge connecting two neighbors of a collapsed vertex creates no new interior vertices:



Thus,  $x$  must also generate elements in  $S'(v, n)$ .

$\therefore$  Every element of  $S'(v, n+1)$  generates an element of  $S'(v, n)$ . This generation can be reversed by removing a non-disconnecting ring edge from the graphs in  $S'(v, n)$  to obtain all the graphs in  $S'(v, n+1)$ .

Lemma 4.9: The graph of a reducer in  $S(v, n)$  has  $3v-3-n$  edges.

Proof: For each  $v$ , column  $v$  begins with the set  $S(v, 2v-2)$  which contains all spanning trees of  $v$  vertices. A spanning tree of a graph of  $v$  vertices has  $v-1$  edges. Thus, for  $n=2v-2$ ,  $v-1 = 3v-3-(2v-2) = 3v-3-n$ . Going down a column, the number of edges is increased by 1; and,  $n$ , the ring size is decreased by 1. Thus, the number of edges in  $S(v, n)$  is  $3v-3-n$ .

Lemma 4.10:  $S_i(v, n) = S(v, n)$  when  $i = 14-n$  and the number of interior vertices,  $m$ , of a reducer in  $S(v, n)$  is less than or equal to  $15-n$ .

Proof: Assume there exists a graph,  $x$ , of a reducer that is in  $S(v, n)$  but not in  $S_i(v, n)$ . From Lemma 4.9, the number of edges in  $x$  is  $3v-3-n$ . The sum of the degrees of the vertices in  $x$  is  $2(3v-3-n)$  since each edge contributes 2 to this sum.

Since  $x \notin S_i(v, n)$ , each uncollapsed ring vertex of  $x$  must have at least degree  $i+1$ .

Let  $c$  be the number of collapsed vertices of  $x$ ;

$u$  be the number of uncollapsed ring vertices of  $x$ ;

$m$  be the number of interior vertices of  $x$ ;

$v = c+u+m$  be the total number of vertices of  $x$ ;

$n$  be the ring size of the reducer associated with  $x$ ;

$e = n - u$  be the number of vertices in the reducer associated with  $x$  that are collapsed vertices of  $x$ .

The sum of the degrees of the vertices of  $x$  must be greater than  $5m + (i+1)u + 2c$ . The summands are justified as follows:

$5m$ : since each interior vertex degree is greater than 4 (Corollary 4.1).

$(i+1)u$ : since each uncollapsed ring vertex degree is  $\geq i+1$  (see definition of  $S_1(v, n)$  and of  $x$ ).

$2c$ : since each collapsed vertex degree is  $\geq 2$  (otherwise, the vertex cannot be collapsed).

$$\therefore 2(3v - 3 - n) \geq 5m + (i+1)u + 2c$$

$$6v - 6 - 2n \geq 5m + (i+1)u + 2c$$

Substitute  $v = c + u + m$

$$6c + 6u + 6m - 6 - 2n \geq 5m + (i+1)u + 2c$$

$$6m - 5m \geq -6c - 6u + 6 + 2n + (i+1)u + 2c$$

$$m \geq 6 + 2n - 6u + (i+1)u + 2c - 6c$$

$$m \geq 6 + (2n - 2u) - 4u + iu + u - 4c$$

$$m \geq 6 + 2(n - u) - 3u + iu - 4c$$

Substitute  $e = n - u$

$$m \geq 6 + 2e + (i-3)u - 4c$$

$$m \geq 6 + (i-3)u + 2e - 4c$$

$e \geq 2c$  since each collapsed vertex in  $x$  comes from collapsing at least 2 vertices of the reducer associated with  $x$ .

$$\therefore 2e - 4c \geq 0$$

$$\therefore m - (2e - 4c) \geq 6 + (i - 3)u$$

Since  $m \geq m - (2e - 4c)$ , rewrite

$$m \geq m - (2e - 4c) \geq 6 + (i - 3)u$$

or

$$m \geq 6 + (i - 3)u$$

Since there must be at least 1 uncollapsed vertex in  $x$ ,  $u \geq 1$ .

$$\therefore m \geq 6 + (i - 3) \cdot 1 = 6 - 3 + i = 3 + i$$

Substitute  $i = 14 - n$

$$\therefore m \geq 3 + 14 - n = 17 - n$$

However,  $m \leq 15 - n$  was assumed.

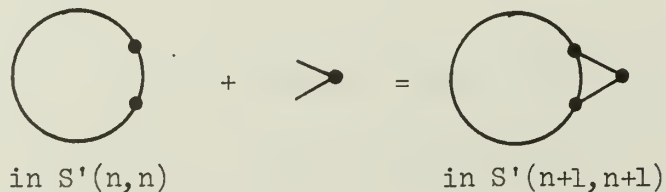
$\therefore x$  cannot exist

$$\therefore S_1(v, n) = S(v, n) \text{ when } i = 14 - n \text{ and } m \leq 15 - n.$$

The modifications to the basic algorithm are described below.

These modifications allowed the sets  $S(v, n)$ , where  $n \leq 11$  and  $v \leq 15$ , to be generated.

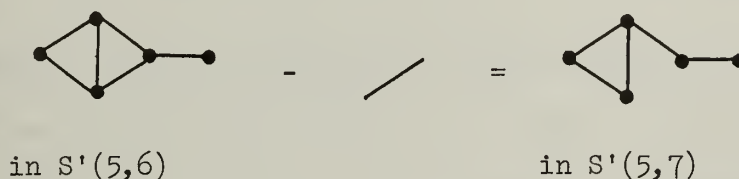
Step 1: The sets,  $S'(n, n)$ , of all  $n$ -ring reducers,  $n \leq 9$ , that have no collapsed or interior vertices were generated.



As shown possible by Lemma 4.4,  $S'(n+1, n+1)$  was generated from  $S'(n, n)$ . A vertex of degree 2 was added to each reducer in  $S'(n, n)$ . When isomorphic reducers were rejected, the resulting set was  $S'(n+1, n+1)$ . The number of elements in  $S'(n, n)$  is calculated in Figure 4.4. The reducers generated in Step 1 formed a subset of the circled entries in Figure 4.3.

Step 2: The sets  $S'(v, n)$  for  $v < n \leq 9$  were generated.

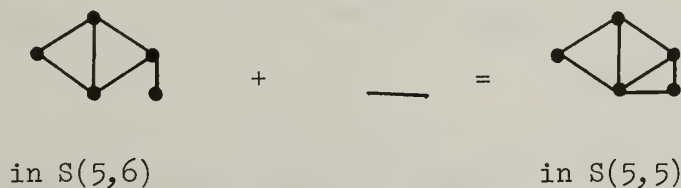
By Lemma 4.8,  $S'(v, n+1)$  can be obtained from  $S'(v, n)$ . Step 2 removed a non-disconnecting ring edge from every reducer in  $S'(v, n)$  to obtain  $S'(v, n+1)$ .



By Lemma 4.5, all the sets that were generated that are located above the dashed line in Figure 4.3 were complete.

Step 3: Generate  $S(v, n)$  from  $S(v, n+1)$ .

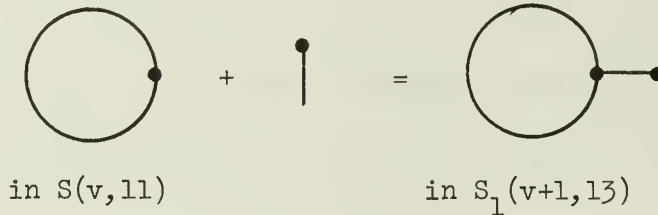
By Lemma 4.6, an edge can be added to all reducers in  $S(v, n)$  to generate  $S(v, n+1)$ .



Step 3 was used to generate  $S(v,n)-S'(v,n)$  for  $v=7,8,9$  and  $n \leq v$  by only generating reducers containing interior vertices. Generation going down the columns of Figure 4.3 was accomplished using Step 3.

Step 4:  $S_1(v+1,13)$  was generated from  $S(v,11)$  for  $v=9,\dots,14$ .

Step 4 added an isthmus (single vertex and single edge, ) adjacent to a ring vertex of a graph in  $S(v,11)$  to obtain a graph in  $S(v+1,13)$ .



Then Step 3 was performed twice to generate  $S_3(v+1,11)$ . By Lemma 4.10,  $S_3(v+1,11) = S(v+1,11)$  when  $m \leq 4$ . Step 4 was used to cross from column  $v$  to column  $v+1$  in Figure 4.3 for  $v=9,10,\dots,14$ . Step 3 was applied to generate the remaining sets in those columns.

Four programs were written to generate the complete set of reducers  $R(n,m)$  where  $6 \leq n \leq 11$  and  $0 \leq m \leq 15-n$ . These programs implemented the steps just described.

Step 1: RING started with  $S'(n,n)$  and went to  $S'(n+1,n+1)$ . A vertex was added that was adjacent to two adjacent ring vertices.

Step 2: REDUP went UP the REDucer column by removing a single edge from  $S'(v,n)$  to go to  $S'(v,n+1)$ .

Step 3: REDDN went Down the REDucer column by adding an edge between two ring vertices that were separated by exactly one other ring vertex.

Step 4: REDCR went aCross the REDucer columns by adding an isthmus to  $S(v, 11)$  to produce  $S_1(v+1, 13)$ .

The algorithms used by the four programs and their upper bound are discussed in the next chapter.

## CHAPTER 5

### IMPLEMENTATION OF REDUCER GENERATION

The four routines to generate the reducers used a common data structure for representing graphs (see Figure 5.1). This structure carried the adjacency information, a code representing the graph and the order of vertices around the ring. The problem was complicated by the fact that a vertex in the graph could have represented several vertices in the reducer. The generation of the reducers was performed by generating the corresponding graphs. (One reducer and its corresponding graph are shown in Figure 4.1.)

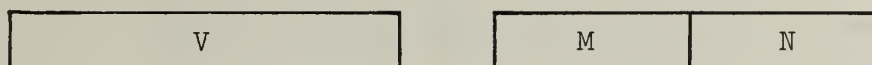
The generation began in Step 1 with the simplest reducers, those with no collapsed or interior vertices. Moving from  $S'(n,n)$  to  $S'(n+1,n+1)$  merely involved adding a vertex between every pair of adjacent ring vertices:



By Lemma 4.4, each of these reducers had to contain a ring vertex of degree 2. Thus, all reducers  $S'(n,n)$ , for  $n=4, \dots, 9$ , were obtained through use of this generation procedure.

The following lemma allowed a reducer in  $S'(n,n)$  to be represented by a code that fits into two words (IBM 360 internal representation).

First two words:

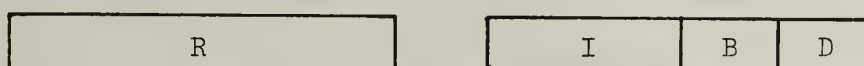


V: number of vertices in the graph (= N when there are no collapsed vertices)

M: number of interior vertices of the graph

N: number of vertices in the ring about the reducer

These two words are followed by V entries of this form:



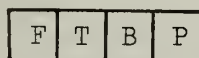
R: The vertex's row from the adjacency matrix (one bit per vertex)

I: Indicator --  $\begin{cases} 1: & \text{interior vertex} \\ \leq 8(v+1): & \text{displacement to the next vertex around the ring} \\ > 8(v+1): & \text{collapsed vertex} \end{cases}$

B: Blank; used for vertex numbering

D: Degree of vertex in the graph

Collapsed vertices point to a chain of halfword entries:



F: Number of 'From' vertex in ring sequence (< 16)

T: Number of 'To' vertex in ring sequence (< 16)

B: Blank; used for vertex number

P: Pointer to next collapsed entry or '0' end of chain

Each reducer requires 60 words:

- 2 Initial words
- 30 Up to 15 vertex entries
- 18 Up to 36 collapsed vertex entries
- 10 Reducer code word

Figure 5.1 Data structure for representing graphs.

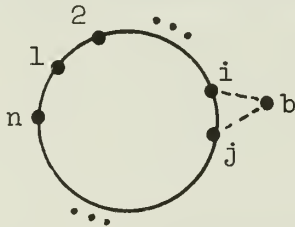
Lemma 5.1: A reducer containing no collapsed or interior vertices is uniquely represented (up to a circular permutation) by listing the degrees of the vertices in order around the ring.

Proof: The proof will proceed by induction on the number of vertices,  $v$ , in the graph of a reducer.

For  $v = 3$ , there is only one graph, . This graph is uniquely represented by '222'.

Assume the lemma is true for  $v \leq n$ . Examine the case when  $v = n+1$ .

Let  $x$  be a reducer with  $n+1$  vertices. By Lemma 4.4,  $x$  has a ring vertex,  $b$ , of degree 2. Remove  $b$  from  $x$  leaving a graph,  $x'$ , of  $n$  vertices. By assumption,  $x'$  has a unique representation  $d_1 d_2 \dots d_i d_j \dots d_n$



$d_i$  = degree of vertex  $i$

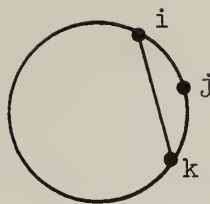
The representation of  $x$  is  $d_1 d_2 \dots (d_i + 1) 2 (d_j + 1) \dots d_n$ . This representation is unique since the representation for  $x'$  was unique and since  $b$ , a degree 2 vertex, is adjacent to vertices  $i$  and  $j$ .

Conversion from the representation to the graph is as follows.

Find a degree 2 vertex,  $j$ , in the representation:

$$d_1 d_2 \dots d_i 2 d_k \dots d_n .$$

$j$  is fixed between  $i$  and  $k$ . The interior of the reducer is completely triangulated; this implies that there must be an edge from  $i$  to  $k$ .



Continue the conversion by finding a degree 2 vertex in the representation  $d_1 d_2 \dots (d_i - 1) (d_k - 1) \dots d_n$ .

The conversion from representation to graph is deterministic. This proves the lemma true for all  $n$ .

Considering the code (representation) as an  $n$ -digit number, each graph could be assigned several codes (due to permutation of the starting position). The largest such code was chosen to represent the graph.

The codes of all non-isomorphic graphs were kept in a binary tree. A binary tree was selected over other search methods, such as sorted or unsorted linear lists. The binary tree method offered search time savings over unsorted linear lists. Keeping a sorted linear list would have involved more program overhead than keeping a binary tree. A comparison of the cost of the binary tree to the cost of linear lists is in Figure 5.2.

Once a graph was generated, the code representing the graph was found. A search was made for the new code in the binary tree containing the codes of previously generated graphs. If the new code was not found, it was inserted into the tree and the data structure of Figure 5.1 was output to disk. If the new code was found in the tree, it was ignored since the graph had been previously generated. In this manner, rejection of isomorphic graphs was accomplished.

| <u>Linear Search</u>      | <u>Binary Tree</u>        |
|---------------------------|---------------------------|
| 1. Load code from pointer | 1. Load code from pointer |
| 2. Test equality          | 2. Test                   |
| 3. Branch equal           | 3. Branch if high         |
| 4. Increment pointer      | 4. Branch if equal        |
| 5. Branch on count to 1   | 5. Load low pointer       |
|                           | 6. Branch to 1            |

| Inner Loop Code      |                     |                                     |                               |
|----------------------|---------------------|-------------------------------------|-------------------------------|
| <u>Reducer</u>       | <u>Size of Code</u> | <u>Left/Right<br/>Pointers Size</u> | <u>% Storage<br/>Increase</u> |
| No interior vertices | 2 words             | 1 word                              | 33%                           |
| 10 vertices          | 5 words             | 1 word                              | 17%                           |
| 15 vertices          | 9 words             | 1 word                              | 11%                           |

Figure 5.2 Storage overhead increases of binary tree method over storing only the codes.

Note that a graph,  $g$ , in  $S(v,n)$  could not be isomorphic to any graph,  $h$ , in  $S(i,j)$ , where  $i \neq v$  or  $j \neq n$ .  $g$  differed from  $h$  in the number of vertices or the number of edges or both. Thus, in Steps 1-4, the isomorphism check was performed only between graphs in the same set,  $S(v,n)$ . Step 1 was finished when the set  $S'(n,n)$ , for  $n=4, \dots, 9$ , had been generated.

Step 2 involved removing a non-disconnecting ring edge from a reducer. This removal created a reducer with a greater ring size. By removing a ring edge from an element in  $S'(v,n)$ , an element of  $S'(v,n+1)$  was obtained. From the circled entries in Figure 4.3, Step 2 was applied to generate  $S'(v,i)$ , for  $i < v$ . By Lemmas 4.5 and 4.8, the sets above the dashed line in Figure 4.3 were complete sets.

Step 3 involved adding an edge between two ring vertices separated by one other ring vertex. The middle vertex was either a collapsed vertex or a vertex of degree greater than 4 (Corollary 4.1). Performing this step involved moving down the columns of Figure 4.3. Since these reducers could have interior vertices, a new method of encoding the graph was developed from L. Weinberg's algorithm [23] for encoding a planar triply-connected graph.

Graphs of reducers were not triply-connected, but reducers were constrained by the fact that they were surrounded by a ring of  $n$  vertices. In addition, the interiors of reducers were completely triangulated. Thus, there were only two ways to embed the graph of a reducer in the plane. These two embeddings were mirror images of each other.

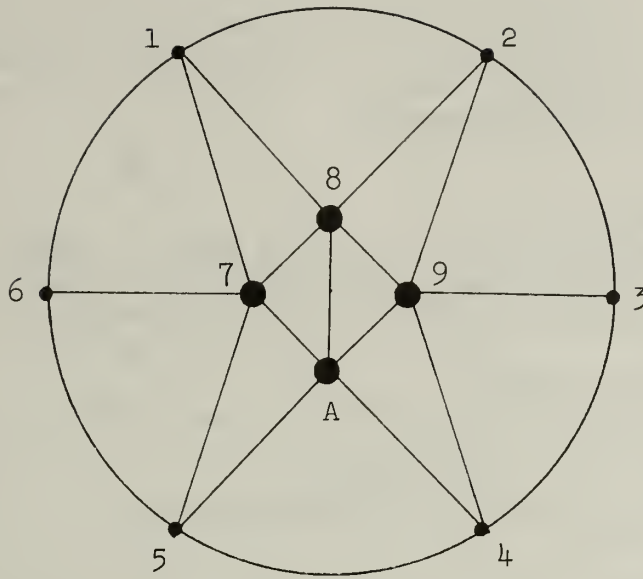
Weinberg's algorithm used a Depth-First Search to traverse all edges of a graph. Each edge was traversed twice (once in each direction). The code was a record of each edge traversed and the direction of traversal. A pair of vertex numbers,  $ij$ , in the code represented the edge between vertices  $i$  and  $j$ , traversed from vertex  $i$  to vertex  $j$ .

The first edge traversed by the modified algorithm (A3) was a ring edge. By starting the code with a ring edge, a circuit of ring edges was visited before visiting any interior edges. Given such a code, there was an easy method to check whether a reducer was also a configuration (A reducer with no collapsed vertices or diagonals was a configuration.) A code beginning with  $2\ 3\ \dots\ n\ 1\ n\ n+1$  and ending with  $n+1\ n\ \dots\ 2\ 1$  contained no collapsed vertices (see Figure 5.3(a)). Graphs of two reducers and their codes are shown in Figure 5.3.

Algorithm A3 required a clockwise ordering of the adjacency list for each vertex. A method was developed for creating this ordered listing by using the adjacency matrix of a graph and the order of the vertices around the ring. An adjacency matrix of a graph is a matrix whose entries,  $a_{ij}$ , are defined as follows:

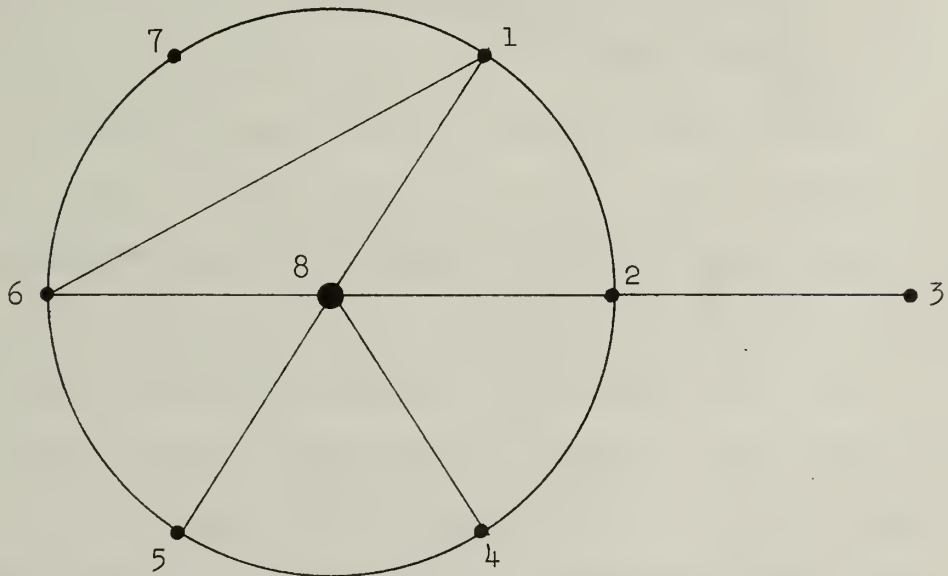
$$a_{ij} = \begin{cases} 1, & \text{if vertex } i \text{ is adjacent to vertex } j \\ 0, & \text{if } i=j \text{ or if vertex } i \text{ is not adjacent to} \\ & \text{vertex } j \end{cases}$$

Row  $i$  of an adjacency matrix is the adjacency row for vertex  $i$ .



Code: 23456167 17818289 293949A4 A5A7A987 57654321

(a) Reducer that is a configuration



Code: 23245671 76168182 84858654 21

(b) Reducer with a collapsed vertex

Figure 5.3 Graphs of two reducers and their codes.

Each vertex,  $i$ , has  $ADJ(i)$  which is a circular listing of the vertices adjacent to  $i$  in clockwise order. The new number assigned to a vertex is  $NEW(i)$ . The graph has  $v$  vertices.  $EDGE(i,j)$  is a binary indicator, 1 if edge  $i$  to  $j$  has been traversed.

Step A (Initialize)

Choose an initial ring vertex,  $i$ , and the next vertex,  $j$ , around the ring from  $i$ . Set  $NEW(i) \leftarrow 0$  for  $k=1, \dots, v$   
 $NEW(i) \leftarrow 1$ ,  $NUM \leftarrow 2$ ,  $EDGE(i,j) \leftarrow 0$  for every  $i,j$ .

Step B (Old or new vertex?)

Set  $EDGE(i,j) \leftarrow 1$ . If  $NEW(j) = 0$ , go to Step C.  
 Output  $NEW(j)$  to the code.  
 If  $EDGE(j,i) = 1$ , go to Step D.  
 Set  $k \leftarrow i$ ,  $i \leftarrow j$ ,  $j \leftarrow k$  (interchange  $i,j$ ).  
 Repeat Step B.

Step C (New vertex)

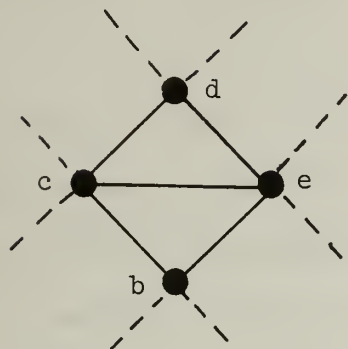
Set  $NEW(j) \leftarrow NUM$ ,  $NUM \leftarrow NUM+1$  and output  $NEW(j)$  to the code.

Step D (Find untraversed edge)

Find  $i$  in  $ADJ(j)$ . Set  $i \leftarrow j$ ,  $j \leftarrow$  next entry in  $ADJ(j)$ .  
 If  $EDGE(i,j) = 0$ , go to Step B.  
 If  $i=1$ ,  $j=2$ , halt.  
 Repeat Step D.

Algorithm A3 Generate code for a reducer.

The method for creating the listing is shown in the following subgraph and portion of an adjacency matrix:



Adjacency matrix

|   | b | c | d | e |
|---|---|---|---|---|
| b | 0 | 1 | 0 | 1 |
| c | 1 | 0 | 1 | 1 |
| d | 0 | 1 | 0 | 0 |
| e | 1 | 1 | 1 | 0 |

The listing of c's adjacent vertices begins with d,e. Taking the logical conjunction (AND) of rows c and e produces two possibilities, b and d, for the next entry in the listing. Since d begins the listing, b is chosen as the next vertex in clockwise order. The listing of the vertices adjacent to c in clockwise order is d,e,b.

The logical conjunction (AND) of two rows of the adjacency matrix could have produced at most 2 possibilities for the next entry in the listing. (This is a consequence of the facts that interiors were completely triangulated and interior vertices had degree greater than 4.) Thus, two initial vertices of the adjacency list had to be known before the remainder of the list could be created. For a ring vertex, the adjacency list began with the next vertex (in clockwise order) around the ring. These two ring vertices could have both been adjacent to a single vertex. If both vertices were adjacent to a single vertex, this single vertex was then the second vertex in the adjacency list. For an interior vertex, the initial two vertices of the adjacency list were found while generating the lists for the other vertices. By starting with ring vertices and spiraling inward, the adjacency lists for all vertices were generated.

Algorithm A3 produced a code that could have started at any ring vertex of a reducer. The lexicographically largest code was chosen to represent the reducer in a binary tree. The length of the code was dependent on the number of edges in the graph. For a given number of vertices,  $v$ , and ring size,  $n$ , the number of edges in the graph was  $3(v-1)-n$ . Since  $v$  was less than 16, the vertex numbers were represented by 4 bits (1 byte). Since each edge required two vertex numbers, the number of words required to encode a reducer was

$$\left\lceil \frac{3(v-1)-n}{2} \right\rceil.$$

The area required to store all the codes of generated reducers in a binary tree is shown in Figure 5.4. Enough core storage was available on the IBM 360/75 to keep the whole tree in core for Step 3.

For Step 4, it was necessary to store part of the tree on disk storage. The tree was allowed to grow until the available core storage was exhausted. Then the remaining tree entries were placed on disk storage. However, only leaves of the tree were placed on disk. Since the root and intermediate nodes of the tree were kept in core, the number of disk reads was kept to a minimum.

In Step 4, the algorithm proceeded from  $S(v,11)$  to  $S_1(v+1,13)$  by adding an isthmus (single vertex and an edge) to each ring vertex of reducers in  $S(v,11)$ . After two applications of Step 3 to  $S_1(v+1,13)$ , the set  $S_3(v+1,11) = S(v+1,11)$  was obtained (Lemma 4.10). A significant aspect of Step 4 was that the generation of the reducers above the double line in Figure 4.1 was not required. Step 4 used the same data structures as Steps 1-3 and also used the isomorphism check of Step 3.

| <u>n,</u><br><u>ring</u><br><u>size</u> | <u>number of vertices, v</u> |                 |                 |                  |
|-----------------------------------------|------------------------------|-----------------|-----------------|------------------|
|                                         | <u>12</u>                    | <u>13</u>       | <u>14</u>       | <u>15</u>        |
| 13                                      | 9 min.<br>250K               | 17 min.<br>444K | 33 min.<br>675K | 74 min.<br>1130K |
| 12                                      | 9 min.<br>263K               | 19 min.<br>563K | 35 min.<br>541K | 80 min.<br>1070K |
| 11                                      | 4 min.<br>116K               | 8 min.<br>212K  | 15 min.<br>196K | 37 min.<br>495K  |
| 10                                      | 1 min.<br>38K                | 2 min.<br>65K   | 5 min.<br>72K   | 13 min.<br>179K  |
| 9                                       | 22 sec.<br>10K               | 45 sec.<br>20K  | 1.5 min.<br>24K | 5 min.<br>76K    |
| 8                                       | 6 sec.<br>3K                 | 14 sec.<br>6K   | 32 sec.<br>9K   | 2 min.<br>32K    |
| 7                                       | 1 sec.<br>1K                 | 5 sec.<br>2K    | 14 sec.<br>3K   | 40 sec.<br>11K   |

Figure 5.4 CPU time and core storage (in K = 1,024 bytes) requirements to generate  $S(v, n)$ .

The number of reducers in  $S(v,n)$  for  $4 \leq v \leq n$  and  $6 \leq n \leq 13$  is shown in Figure 5.5. The number in parenthesis is the number of reducers if isomorphic reducers are not rejected.

The generation of all the reducers was broken into steps, each of which could be run as a single program on the 360/75 as many times as necessary. Although the programs used large amounts of core storage (up to 1 million bytes) and disk storage (13 million bytes for the largest file; 18 million total bytes), the factor that limited further generation was the amount of CPU time allowed for one program. The longest program run (80 CPU minutes) translated to about 4 hours real time. This was close to the mean time between two system errors of the 360/75 (when all programs must be restarted). In addition, generation of graphs with 16 or more vertices would have required modification of the data structures.

|              |   | number of vertices, v |   |   |   |   |   |    |    |    |    |    |    |
|--------------|---|-----------------------|---|---|---|---|---|----|----|----|----|----|----|
|              |   | 4                     | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| ring size, n | 6 |                       |   |   |   |   |   |    |    |    |    |    |    |
|              | 7 |                       |   |   |   |   |   |    |    |    |    |    |    |
| 8            |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 9            |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 10           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 11           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 12           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 13           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 14           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 15           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 16           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 17           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 18           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 19           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 20           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 21           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 22           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 23           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 24           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 25           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 26           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 27           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 28           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 29           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 30           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 31           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 32           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 33           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 34           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 35           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 36           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 37           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 38           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 39           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 40           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 41           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 42           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 43           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 44           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 45           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 46           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 47           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 48           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 49           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 50           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 51           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 52           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 53           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 54           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 55           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 56           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 57           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 58           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 59           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 60           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 61           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 62           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 63           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 64           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 65           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 66           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 67           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 68           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 69           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 70           |   |                       |   |   |   |   |   |    |    |    |    |    |    |
| 71           |   |                       |   |   |   |   |   |    |    |    |    |    |    |

(The number of reducers generated before isomorph rejection is shown in parentheses)

Figure 5.5 Number of reducers in  $S(v,n)$ .

## CHAPTER 6

## C-REDUCTION

Once the reducers were generated, they had to be prepared for use in C-reduction. In this chapter, C-reduction and the generation of the direct colorations of the reducers are discussed. The generation of the direct colorations of the reducers was done in a manner similar to the generation of the direct coloration of configurations. Instead of an adjacency list, the reducers were input using the data structure of Figure 5.1. Also, the output was in a different format.

Two decisions were made that minimized the disk storage space needed to store the direct colorations of the reducers. The first choice was to store a bit string with one bit per possible coloration. The comparison of a bit string to the ring coloration representation described in Chapter 3 is shown in Figure 6.1. The rapid look-up (hashing) feature that was useful in determining the direct and indirect colorations of configurations was no longer needed. A configuration,  $B$ , is C-reduced by a reducer,  $R$ , whenever the direct set of colorations of  $R$  is a subset of the good colorations of  $B$ . A bit string was well suited to checking the condition  $D(R) \subseteq G(B)$ . The second decision involved storing only 1 bit string per reducer. The ring of the reducer had to be matched to the ring of the configuration in every possible way. Selecting one of the  $n$  vertices as the initial vertex and reading the remaining ring vertices in clockwise or counterclockwise order, yields  $2n$  possible distinct ways

| <u>ring size</u> | <u>number of bytes<br/>required for bit<br/>list area</u> | <u>number of bytes<br/>required using 1 bit<br/>per coloration</u> |
|------------------|-----------------------------------------------------------|--------------------------------------------------------------------|
| 4                | 1                                                         | 1                                                                  |
| 5                | 2                                                         | 2                                                                  |
| 6                | 11                                                        | 4                                                                  |
| 7                | 42                                                        | 12                                                                 |
| 8                | 171                                                       | 35                                                                 |
| 9                | 682                                                       | 103                                                                |
| 10               | 2,731                                                     | 308                                                                |
| 11               | 10,922                                                    | 923                                                                |
| 12               | 43,691                                                    | 2,768                                                              |
| 13               | 174,762                                                   | 8,304                                                              |
| 14               | 699,051                                                   | 24,912                                                             |
| 15               | 2,796,202                                                 | 74,734                                                             |
| 16               | 10,485,760                                                | 224,202                                                            |

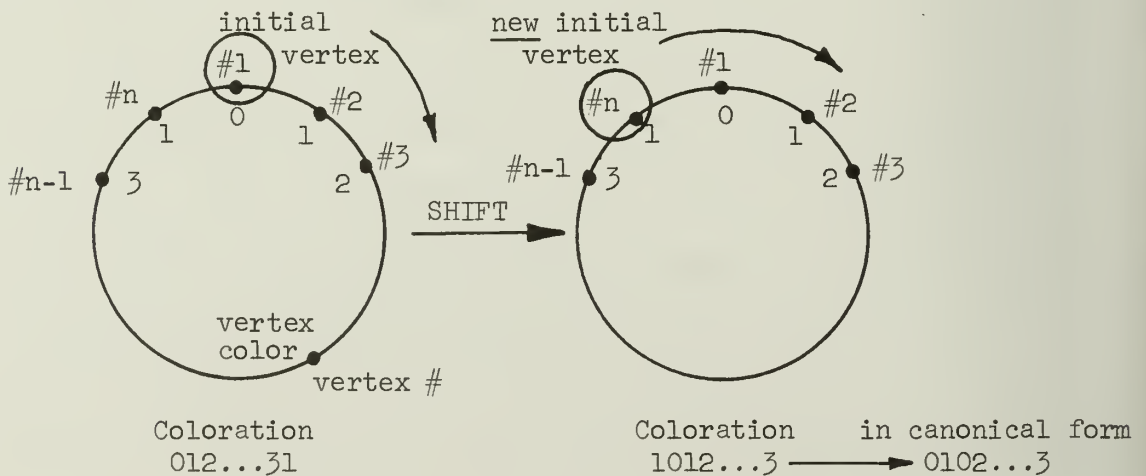
Figure 6.1 Comparison of bit list area  
and 1 bit per coloration schemes.

to match reducer to configuration. The decision was made to fix the initial vertex and ordering of the colorations of each reducer, and the colorations of the configuration were to be shifted. This decision allowed only 1 bit string to be stored per reducer; however, each good coloration of a configuration had to be listed in  $2n$  ways. In this manner, disk storage space and disk reads were saved.

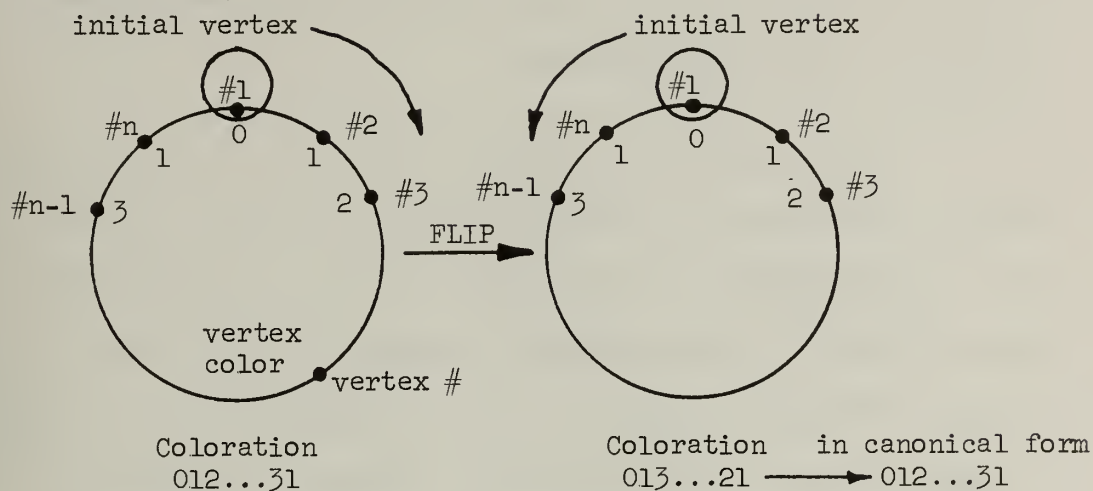
Given a ring coloration, it is a simple matter to choose a new initial vertex and ordering. The only difficulty arises when color interchanges have to be performed to restore the coloration to canonical form. This procedure was mechanized to generate  $2n$  bit strings of good colorations from the bit string initially calculated.

The generation of the  $2n$  bit strings was accomplished by using two program-generated routines, SHIFT and FLIP. In these routines, three statements were generated for every possible ring coloration. These statements tested the condition of the bit representing a coloration in the original string and set this condition in the new string.

The SHIFT program changed the initial vertex of a coloration to the next vertex counterclockwise on the ring.



The FLIP program changed the ordering of the coloration from clockwise to counterclockwise:



The program SHIFT was applied  $n-1$  times to the original bit string of the configuration. The program FLIP was applied to the original bit string; and then, the output from FLIP was shifted  $n-1$  times by applying the routine SHIFT. The application of the two routines resulted in  $2n$  bit strings. These  $2n$  bit strings were compared to the single bit string of each reducer.

The program to attempt C-reduction (CRED) was written to allow any of the following procedure choices. For each configuration, CRED could provide the following information if requested:

1. All reducers which reduced the configuration;
2. The first reducer (in the order of generation) which reduced the configuration;
3. A list of the direct colorations of each reducer that were bad colorations of the configuration;

4. The first reducer with the smallest number of direct colorations which were bad colorations of the configuration.

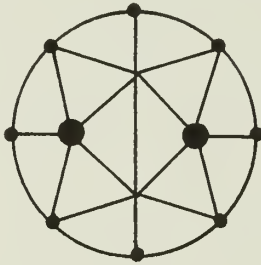
The reducers were sorted by ring size. Within ring size, the reducers were ordered by increasing number of interior vertices. There were three advantages to this ordering. First, it allowed the reducers with the smallest number of direct colorations to be tried first. These reducers seemed more likely to reduce a configuration than reducers with a larger number of direct colorations. Second, this ordering speeded up the removal of redundant reducers. By grouping reducers with the same number of interior vertices, the disk file could be read sequentially which minimized the time required to examine all potential reducers. Third, since the potential reducers were sorted by the number of interior vertices, the first effective reducer found contained the minimum number of interior vertices of any effective reducer of the configuration. The reducers enumerated in Figure 5.5 were sorted by the number of total vertices. Figure 6.2 shows the number of reducers after they were sorted by the number of interior vertices.

After all of the reducers were generated and their direct colorations listed, the redundant reducers were removed. To determine whether a reducer,  $x$ , was redundant, the indirect colorations of  $x$  were generated. Then, C-reduction was attempted using reducers with fewer interior vertices than  $x$ . Since all configurations of ring size  $n$ ,  $6 \leq n \leq 11$  and  $m \leq 15-n$ , were in the list of reducers, one added result of the removal operation was the determination of the reducibility or irreducibility of each such configuration.

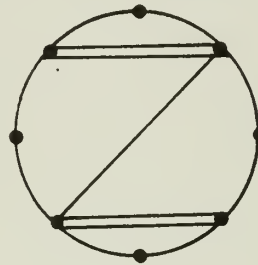
| Number of<br>interior<br>vertices | Ring size |            |            |          |           |           |
|-----------------------------------|-----------|------------|------------|----------|-----------|-----------|
|                                   | <u>6</u>  | <u>7</u>   | <u>8</u>   | <u>9</u> | <u>10</u> | <u>11</u> |
| 0                                 | 8         | 12         | 40         | 95       | 323       | 982       |
| 1                                 | 2         | 7          | 27         | 112      | 463       | 1883      |
| 2                                 | 1         | 4          | 23         | 108      | 544       | 2685      |
| 3                                 | 1         | 4          | 23         | 143      | 820       | 4472      |
| 4                                 | 1         | 5          | 39         | 241      | 1490      | 9332      |
| 5                                 | 1         | 8          | 70         | 483      | 3378      | 4521      |
| 6                                 | 4         | 25         | 188        | 1322     | 1459      |           |
| 7                                 | 9         | 73         | 560        | 494      |           |           |
| 8                                 | <u>39</u> | <u>287</u> | <u>216</u> | —        | —         | —         |
| Total                             | 66        | 425        | 1186       | 2998     | 8477      | 23875     |

Figure 6.2 Number of reducers generated for each ring size and number of interior vertices.

Since all potential reducers were generated, all the effective reducers of a configuration having ring size  $6 \leq n \leq 11$  and  $m \leq 15-n$  could be examined. For instance, it was discovered that there is only one effective reducer of the following configuration:



Configuration



Reducer

This effective reducer was found by A. Bernhart [4].

The following definitions are used in the subsequent conjecture.

A configuration is geographically very good if:

1. No interior vertex of the configuration has 3 non-consecutive adjacent vertices on the ring.

and

2. No interior vertex of the configuration has the property that its removal splits the interior vertices into components one of which has exactly two vertices.

A configuration is acceptable if it contains a geographically very good sub-configuration.

Conjecture: All reducible configurations are acceptable configurations.

In the process of eliminating redundant reducers, this conjecture was shown true for configurations with ring size  $6 \leq n \leq 11$  having  $m$  interior vertices,  $m \leq 15-n$ .

An attempt was made to disprove the conjecture by finding a configuration that was not acceptable but was reducible. A list of the attempted configurations is in Appendix A. No sub-configuration of any configuration attempted was geographically very good. The configurations in Appendix A contain  $m$  interior vertices where  $m$  is less than or equal to the ring size,  $n$ . Since it has been observed that an acceptable configuration with  $m \geq n-2$  is almost certain to be reducible, the unacceptable configurations used in the attempt were chosen to satisfy this condition.

One of the most important results of this thesis work is a list of reducible configurations which have no effective reducers without interior vertices. Appendix B contains four configurations and the first effective reducers found by CRED. (Although the 10-ring configuration was previously reduced by F. Allaire and E. Swart [1], it was not known if the configuration had any effective reducers with no interior vertices.) Each effective reducer in Appendix C contains more than one interior vertex; therefore, all reducers with no interior vertices fail for these configurations. This conclusion is a result of the previously mentioned order of search through the reducers. Hence, the use of only reducers without interior vertices is not sufficient to reduce all reducible configurations.

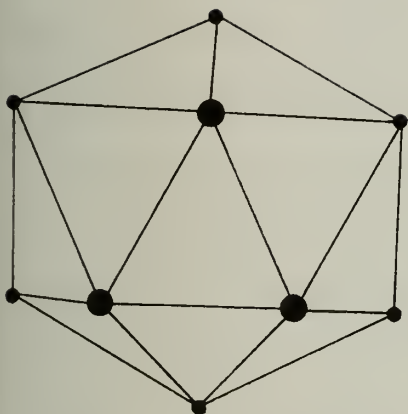
## CHAPTER 7

### PROGRAM OUTPUT

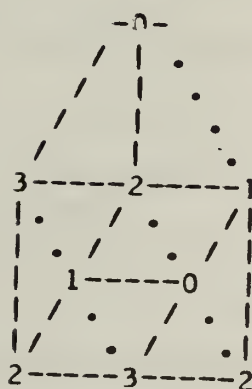
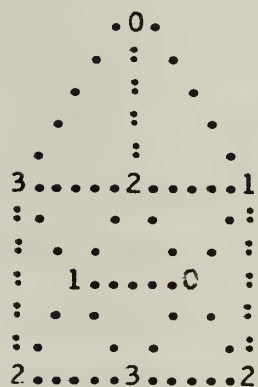
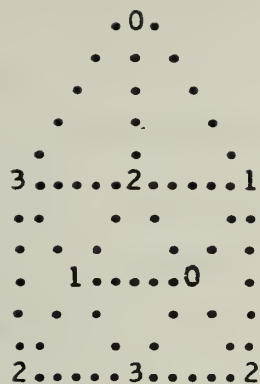
Demonstration of the reducibility or irreducibility of configurations with large ring sizes involves a large amount of output. This chapter is an examination of the output used to initially check each program and a discussion of how large a truly "hand-checkable" proof might be.

The program DIRECT produced the direct colorations of a configuration. Initially, a list of the direct colorations and of the corresponding colors of the configuration's interior vertices was used to verify that DIRECT was correct. Such a list for an 11-ring configuration would involve listing between 30%-50% of the 7,381 possible colorations of an 11-ring (between 2,214 and 3,690 colorations). One method to speed up the checking of the direct colorations that was attempted used the following idea: A planar representation of the configuration with each vertex replaced by its color can be rapidly checked visually (see Figure 7.1). Thus, two adjacent vertices that have the same color are easily spotted; and, an entire page of direct colorations in picture format is easier to check visually than a list of the colors of the vertices.

The picture format, however, required more pages of output than a list of the direct colorations. In addition, the picture format was not easily automated. New formats had to be created for each new configuration.



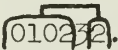
Configuration  
(5-5-5 triangle in  
a 6-ring)



Printouts

Figure 7.1 Example formats of coloration printouts.

An option in DIRECT allowed a list of the direct colorations of a configuration to be printed, but such lists were not generally produced because of their size.

The indirect colorations, a product of the program DRED, required more computation (and thus more output to check) than the direct colorations. D-reduction involved the determination of the color pair segments of a coloration and color pair interchanges within segments (described in Chapter 3). An example of the output used to depict the operations performed by DRED is shown in Figure 7.2. Each of the three possible color pairings is followed by the columns determined by that color pairing. Within each column, the colorations produced by interchanges are listed. These colorations are followed by a pictorial representation of the color interchanges that were performed on the original coloration. For instance, column 2 of color pair 02,13 in Figure 7.2 shows the following: "01-0212 --\*-" . The segment structure is . The asterisk (\*) indicates the position in the four rightmost positions that had a color interchange. (Recall that segments connected to the first two vertices, 01, were never used in color pair interchanges.) A dash (-) shows each position that remained unchanged. To the right of the listing of each good coloration the word "GOOD" was printed. Nothing was printed for a bad coloration. In some cases, it was necessary to interchange the colors 2 and 3 throughout the entire coloration to return the coloration to canonical form. The phrase "2-3 INT" was printed whenever such an interchange was necessary. The listing of the coloration printed was the listing after this 2-3 interchange had been performed. When at least one good coloration had been found in every column of a

```

*****
* COLORATION 01- 0232      *
*****

                                COLOR PAIR (01,23):
                                COLOR PAIR (02,13):

COLUMN 1
  01- 2032  **--
COLUMN 2
  01- 0212  --*-
                                COLOR PAIR (03,12):
COLUMN 1
  01- 2323  *---  2-3 INT  GOOD
COLUMN 2
  01- 0123  -*--  2-3 INT  GOOD
COLUMN 3
  01- 2303  *-*-  2-3 INT  GOOD
COLUMN 4
  01- 0202  --*-          GOOD
COLUMN 5
  01- 2323  *---  2-3 INT  GOOD

COLORATION 01- 0232      MADE GOOD

```

Figure 7.2 D-reduction sample printout.

color pairing, the original coloration had been made good and the phrase printed was "COLORATION 01-XXXX MADE GOOD".

The output presented in the format shown in Figure 7.2 was used to verify the operations of both the single pass (DRED) and multiple pass (DREDP) D-reduction programs. This format clearly displayed the operations involved in D-reduction and allowed the operations to be checked by hand. However, this format created a large amount of output for each configuration. The matrices of colorations that remained bad after DREDP were printed each time a pass was made through the coloration list. The multiple printing of matrices was eliminated in DRED by using only a single pass through the coloration list. In DRED, if one coloration became good it could cause other colorations to become good. Thus, it was sometimes necessary to examine the matrices between the time D-reduction was attempted for a certain coloration and the time that coloration became good. While the operation of D-reduction could be checked using the output format of Figure 7.2, the amount of output produced was large and varied greatly between configurations. The amount of output decreased as the number of good colorations increased. However, for a 9-ring configuration with 224 direct colorations and 218 indirect colorations (out of a possible 820 colorations), 800 pages of output were produced. Instead of this large output, counts of the direct and indirect colorations were used as rapid checks of D-reduction. These two counts were compared with the counts of previously reduced configurations and D-reduction was correct if the counts were equal.

For  $R$  to be an effective reducer of a configuration,  $W$ ,  $D(R)$  had to be a subset of  $G(W)$  (the union of  $D(W)$  and  $I(W)$ ). To verify

C-reduction, the direct colorations of  $R$  had to be verified as well as the direct and indirect colorations of  $W$ .  $G(W)$  was verified by checking the output of DIRECT and DRED after they were applied to the configuration,  $W$ . An option of the C-reduction program (CRED) permitted the printing of the direct colorations of an effective reducer. In most cases, the output from CRED was limited instead to listing the encoding of an effective reducer. Along with the encoding, the initial position of the colorations of the configuration was indicated by printing the flipped and shifted conditions of the colorations. For example,

CONFIGURATION C-REDUCED BY #5648

CONFIGURATION WITH 2 SHIFTS AND FLIPPED;

|          |          |          |
|----------|----------|----------|
| 23456789 | 19A1AB1B | 5B6BC6C7 |
| C8CACBA8 | A9876515 | 41431321 |
| D1000000 |          |          |

Proving the irreducibility of a configuration,  $W$ , involved much more work and output than proving reducibility. The outputs from DIRECT and DRED applied to  $W$  were again checked. An option in CRED permitted the printing, for each reducer  $R$ , the colorations in  $D(R)$  that were not in  $G(W)$ . Even when only one such coloration was printed for each reducer, the output produced was large. For an 11-ring configuration containing 5 interior vertices, it would have required listing 22 colorations for each of the 24,395 reducers. One coloration is required for each of the  $2 \cdot n$  ( $2 \cdot 11 = 22$ ) possible shifted and/or flipped positions of the configuration. (The number of positions could be somewhat decreased

by eliminating isomorphic positions of the configuration.)

Irreducibility proofs for configurations with rings of 9 or more vertices involved so much computation as to make hand verification impractical. For configurations of this size the only practical check of irreducibility is an independently written reduction program. Indeed, even reducibility of configurations with rings of 12 or more vertices cannot be easily checked by hand because of the large number of possible ring colorations ( $> 22,000$ ; see Figure 7.3). Thus, while the options were written into DIRECT, DRED and CRED to print hand-checkable proofs, the sheer size of the output prevented the printing of such proofs.

No effective reducers were found for any of the 9-ring configurations listed in Appendix D. These configurations were listed in Table 2 [7] and Frank Bernhart had conjectured that these configurations were irreducible. Along with each configuration in Appendix D is the number of direct and indirect colorations of the configurations.

| <u>ring size</u> | <u>number of colorations</u> |
|------------------|------------------------------|
| 6                | 31                           |
| 7                | 91                           |
| 8                | 274                          |
| 9                | 820                          |
| 10               | 2461                         |
| 11               | 7381                         |
| 12               | 22144                        |
| 13               | 66430                        |
| 14               | 199291                       |
| 15               | 597871                       |
| 16               | 1793614                      |

Figure 7.3 Total number of possible colorations of an n-ring.

## CHAPTER 8

## CONCLUSIONS AND EXTENSIONS

The Four Color Conjecture would be proven true if an unavoidable set of reducible configurations was found. The study of reducibility for configurations having ring size  $6 \leq n \leq 11$  and  $m \leq 15-n$  interior vertices using a complete set of potential reducers has important implications for reductions of larger configurations ( $n > 11$ ). First, reducers containing interior vertices are necessary for reduction (Chapter 6). Second, acceptable configurations with  $n > 11$  are likely to be reducible (Conjecture from Chapter 6). Third, the configurations that were reduced (Appendices B and C) are useful since all larger configurations which contain them as sub-configurations are also reducible. Finally, many of the techniques used to reduce configurations with ring size  $n \leq 11$  can be extended to configurations with larger ring size.

My work also allowed other reducibility results to be verified. After this work was nearly complete, the work of F. Allaire and E. Swart (to appear [1]) to set up a comprehensive list of reducible configurations up to ring size 10 was brought to my attention. All of their reducibility results were confirmed; in addition, the 10-ring configuration in Appendix C was shown to have no effective reducers without interior vertices. All of the configurations that Allaire and Swart listed as 'NR' (not reducible) were shown to be irreducible (if  $m \leq 16-n$ ) or partially irreducible ( $m > 16-n$ ).

The programs DIRECT and DRED were written so they could be extended to configurations with  $n > 11$ . K. Appel [3] has extended these programs to configurations with  $n = 12, 13, 14$ . The ring coloration representation of Chapter 3 was used by Appel even though it required a bit list area of 699,051 bytes for  $n = 14$ . As was mentioned in Chapter 3, other schemes exist that can reduce the size of the bit list area; but, such schemes increase the time required to determine if a coloration is good. Since the number of possible ring colorations increases by a factor of 3 for each ring vertex added (see Figure 7.3), D-reduction above  $n = 16$  is not considered practical at present. The speed and core storage of present computers is simply not adequate.

My technique for C-reduction using a complete set of reducers whose direct colorations are stored on disk cannot easily be extended to  $n = 12$  or more. Such an extension would involve a significant expansion in the required disk storage space. The size of the disk storage space required can be lower bounded by considering the reducers in  $S(v, 12)$  that were used to generate  $S_3(v, 11)$  (Chapter 5). Each 12-ring reducer would require 2,768 bytes to store its direct colorations. Thus, 170 million bytes or approximately 170 removable IBM 2314 disk packs would be required to store the direct colorations of the 62,831 reducers in  $S_2(v, 12)$  (see Figure 5.5). Computation of irreducibility for configurations with ring sizes  $n > 11$  appears to be limited by the number of potential reducers that must be examined.

It will be mentioned now that A. Bernhart [5] has discovered a technique that sometimes increases the number of good colorations

beyond direct and indirect colorations. However, no irreducible configuration has been found that becomes reducible when the increased number of good colorations was used. In fact, the two configurations in Appendix E remained irreducible even when the extra good colorations (calculated by F. Bernhart [6]) had been added.

Further research can be done to determine the reducers most likely to reduce a configuration. Since only one effective reducer is needed out of the huge number of potential reducers when  $n > 11$ , more configurations could be reduced if such research were successful. Many heuristics are now being used to select the configurations most likely to be reducible but not to select the reducers most likely to reduce a given configuration. Some results on the selection of such reducers could come from the work on Splicing Theory and open sets [22]. In fact, the reducers of the configurations in Appendix C do conform to the Splicing Theory. However, this theory has not yet been developed to the point where it can be used to select effective reducers or even reducers that are most likely to be effective reducers.

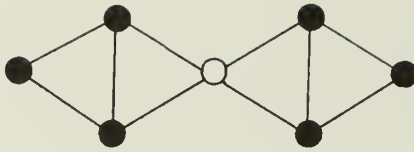
## LIST OF REFERENCES

- [1] F. Allaire and E. R. Swart, "A systematic approach to the determination of reducible configurations in the four-color conjecture," to appear in J. Comb. Theory.
- [2] K. I. Appel and W. Haken, "The existence of unavoidable sets of geographically good configurations," preprint.
- [3] K. I. Appel, W. Haken and J. A. Koch, "Reducible configurations with ring sizes up to 14 vertices," in preparation.
- [4] A. F. Bernhart, "Another reducible edge configuration," Amer. J. Math. 70 (1948), 144-146.
- [5] A. F. Bernhart, "Irreducible six-rings in minimal five-color maps," Amer. J. Math. 69 (1947), 391-412.
- [6] F. R. Bernhart, "Topics in graph theory related to the five-color conjecture," Thesis, Kansas State University, 1973.
- [7] F. R. Bernhart, "On the characterization of reductions of small order," preprint.
- [8] G. D. Birkhoff, "The reducibility of maps," Amer. J. Math. 35 (1913), 114-128.
- [9] P. Franklin, "Note on the four color problem," J. Math. Phys. 16 (1938), 172-184.
- [10] F. Guthrie, "Note on the colouring of maps," Proc. Royal Soc. Edinburg 10 (1880), 729.
- [11] B. T. Goldbeck, "8-rings in minimal maps," Thesis, University of Oklahoma, 1957.
- [12] F. Harary, Graph Theory, Addison-Wesley, Reading, Mass., 1969.
- [13] P. J. Heawood, "Map-colour theorem," Quart. J. Math. 24 (1890), 322-338.
- [14] H. Heesch, Untersuchungen zum Vierfarbenproblem, Bibliographisches Institut AG, Mannheim, 1969.
- [15] A. B. Kempe, "On the geographical problem of the four colours," Amer. J. Math. 2 (1879), 193-204.

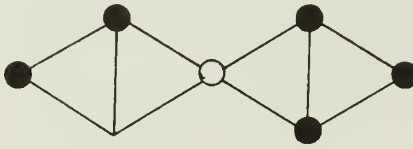
- [16] C. L. Liu, Topics in Combinatorial Mathematics, Mathematical Association of America, 1972.
- [17] J. Mayer, "Probleme des quatres couleurs: Un contre-exemple doit avoir au moins 96 sommets," to appear in J. Comb. Theory.
- [18] O. Ore, The Four Color Problem, Academic Press, New York, 1967.
- [19] O. Ore and J. Stemple, "Numerical calculations on the four color problem," J. Comb. Theory 8 (1970), 65-78.
- [20] G. Ringel and J. W. T. Youngs, "Solution of the Heawood Map Coloring Problem," Proc. Nat. Acad. Sci. 60 (1968), 438-445.
- [21] W. Stromquist, "The four color theorem for small maps," J. Comb. Theory 19 (1975), 256-268.
- [22] W. Stromquist, "The four color conjecture and splicing," Thesis, Harvard University, 1975.
- [23] L. Weinburg, "A simple and efficient algorithm for determining isomorphism of planar triply connected graphs," IEEE Trans. Circuit Theory CT-13, 2 (1966), 142-148.
- [24] C. E. Winn, "On the minimum number of polygons in an irreducible map," Amer. J. Math. 62 (1940), 406-416.

## APPENDIX A

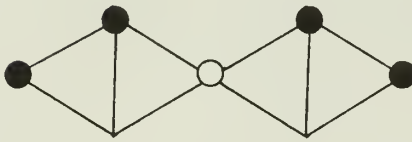
Configurations with ring size 8, 9, 10 or 11 that are not acceptable configurations and that were not reduced.



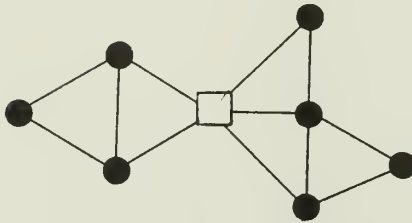
| <u>m</u> | <u>n</u> | <u>direct</u> | <u>indirect</u> |
|----------|----------|---------------|-----------------|
| 7        | 9        | 237           | 267             |



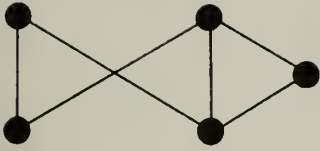
|   |    |     |      |
|---|----|-----|------|
| 7 | 10 | 560 | 1022 |
|---|----|-----|------|



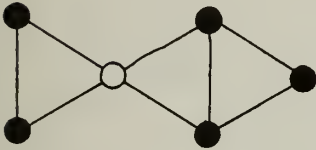
|   |    |      |      |
|---|----|------|------|
| 7 | 11 | 1320 | 3854 |
|---|----|------|------|



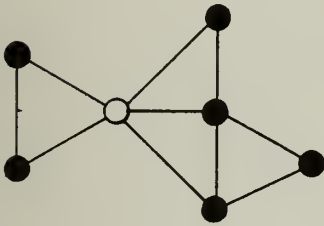
|   |    |     |     |
|---|----|-----|-----|
| 8 | 10 | 608 | 890 |
|---|----|-----|-----|



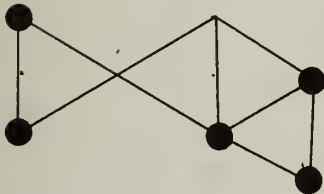
| <u>m</u> | <u>n</u> | <u>direct</u> | <u>indirect</u> |
|----------|----------|---------------|-----------------|
| 6        | 8        | 101           | 99              |



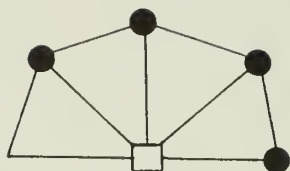
|   |   |     |     |
|---|---|-----|-----|
| 6 | 9 | 237 | 267 |
|---|---|-----|-----|



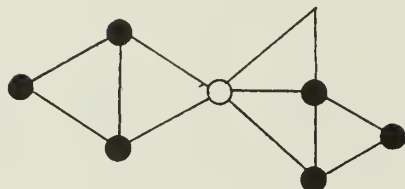
|   |   |     |     |
|---|---|-----|-----|
| 7 | 9 | 265 | 326 |
|---|---|-----|-----|



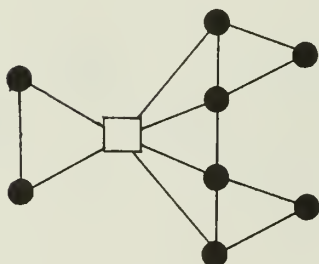
|   |   |     |     |
|---|---|-----|-----|
| 7 | 9 | 268 | 301 |
|---|---|-----|-----|



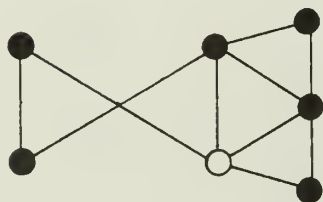
| <u>m</u> | <u>n</u> | <u>direct</u> | <u>indirect</u> |
|----------|----------|---------------|-----------------|
| 6        | 10       | 507           | 1375            |



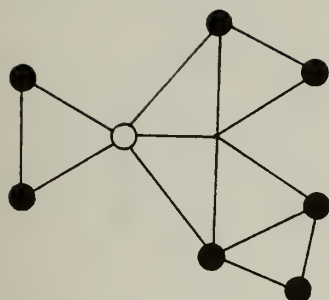
|   |    |     |     |
|---|----|-----|-----|
| 8 | 10 | 594 | 794 |
|---|----|-----|-----|



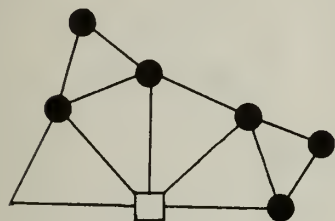
|   |    |     |     |
|---|----|-----|-----|
| 9 | 10 | 703 | 845 |
|---|----|-----|-----|



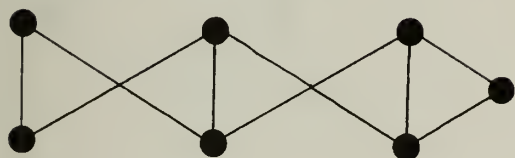
|   |    |     |     |
|---|----|-----|-----|
| 8 | 10 | 657 | 949 |
|---|----|-----|-----|



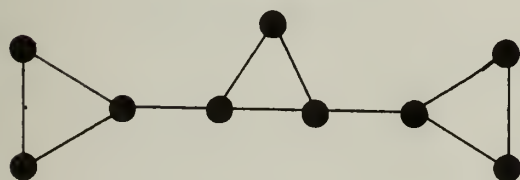
| <u>m</u> | <u>n</u> | <u>direct</u> | <u>indirect</u> |
|----------|----------|---------------|-----------------|
| 9        | 10       | 749           | 814             |



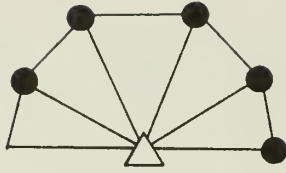
|   |    |     |     |
|---|----|-----|-----|
| 8 | 10 | 616 | 988 |
|---|----|-----|-----|



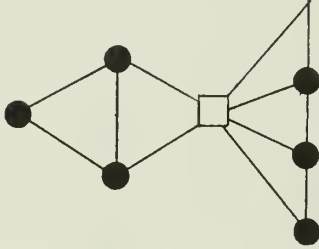
|   |    |     |     |
|---|----|-----|-----|
| 9 | 10 | 709 | 688 |
|---|----|-----|-----|



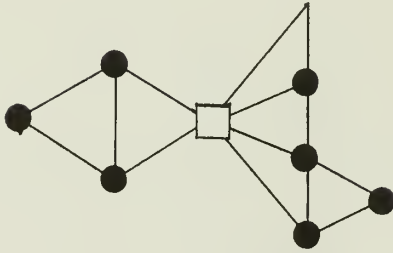
|   |    |     |     |
|---|----|-----|-----|
| 9 | 10 | 648 | 799 |
|---|----|-----|-----|



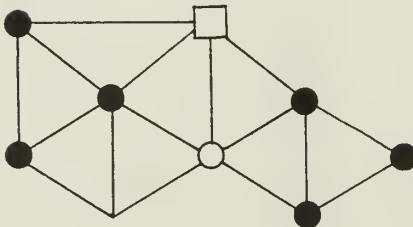
| <u>m</u> | <u>n</u> | <u>direct</u> | <u>indirect</u> |
|----------|----------|---------------|-----------------|
| 7        | 11       | 1257          | 4010            |



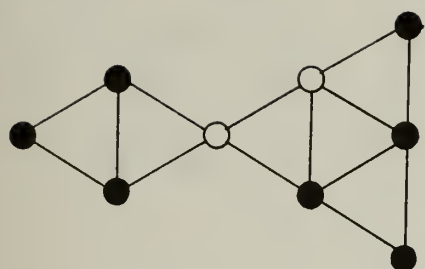
|   |    |      |      |
|---|----|------|------|
| 8 | 11 | 1374 | 2873 |
|---|----|------|------|



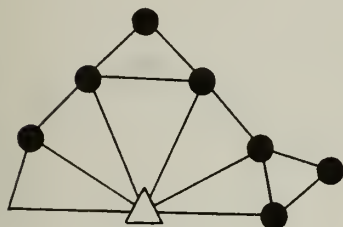
|   |    |      |      |
|---|----|------|------|
| 9 | 11 | 1543 | 2578 |
|---|----|------|------|



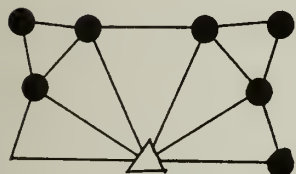
|   |    |      |      |
|---|----|------|------|
| 9 | 11 | 1557 | 2596 |
|---|----|------|------|



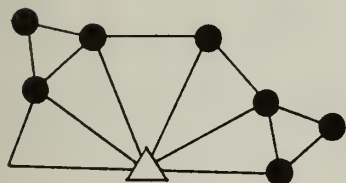
| <u>m</u> | <u>n</u> | <u>direct</u> | <u>indirect</u> |
|----------|----------|---------------|-----------------|
| 9        | 11       | 1517          | 2597            |



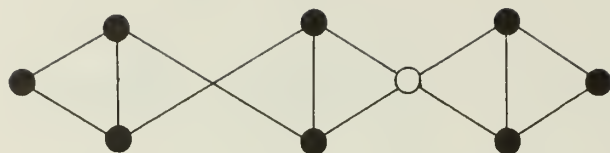
|   |    |      |      |
|---|----|------|------|
| 9 | 11 | 1562 | 3006 |
|---|----|------|------|



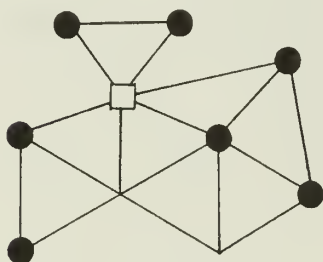
|   |    |      |      |
|---|----|------|------|
| 9 | 11 | 1545 | 3004 |
|---|----|------|------|



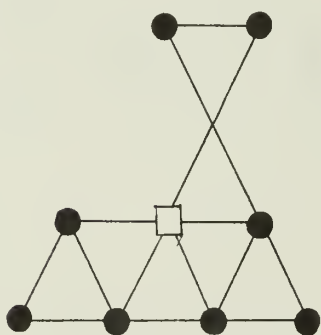
|   |    |      |      |
|---|----|------|------|
| 9 | 11 | 1545 | 3206 |
|---|----|------|------|



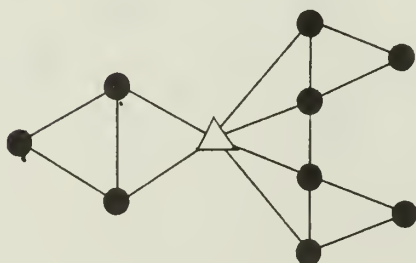
| <u>m</u> | <u>n</u> | <u>direct</u> | <u>indirect</u> |
|----------|----------|---------------|-----------------|
| 10       | 11       | 1690          | 1836            |



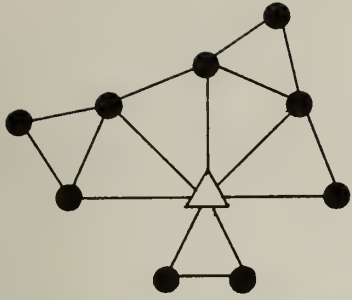
|    |    |      |      |
|----|----|------|------|
| 10 | 11 | 2035 | 2416 |
|----|----|------|------|



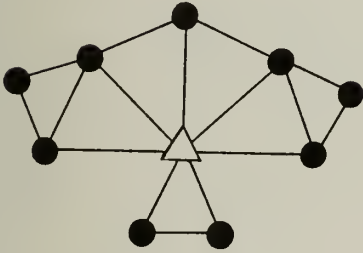
|    |    |      |      |
|----|----|------|------|
| 10 | 11 | 1825 | 2540 |
|----|----|------|------|



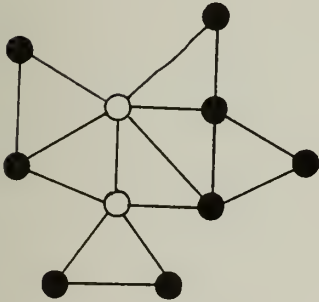
|    |    |      |      |
|----|----|------|------|
| 10 | 11 | 1668 | 2384 |
|----|----|------|------|



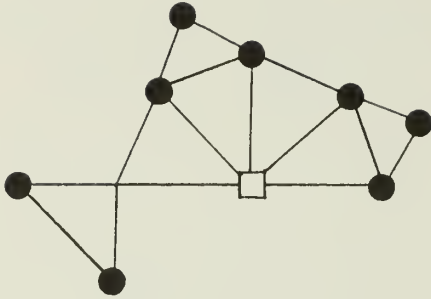
| <u>m</u> | <u>n</u> | <u>direct</u> | <u>indirect</u> |
|----------|----------|---------------|-----------------|
| 10       | 11       | 1838          | 2590            |



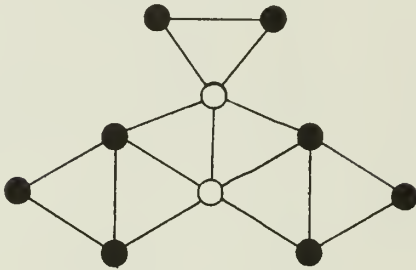
|    |    |      |      |
|----|----|------|------|
| 10 | 11 | 1830 | 2739 |
|----|----|------|------|



|    |    |      |      |
|----|----|------|------|
| 10 | 11 | 1974 | 2521 |
|----|----|------|------|



| <u>m</u> | <u>n</u> | <u>direct</u> | <u>indirect</u> |
|----------|----------|---------------|-----------------|
| 10       | 11       | 1825          | 2540            |

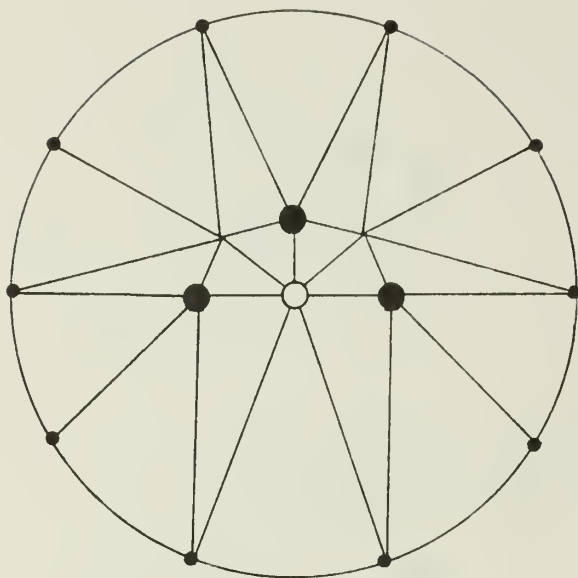


|    |    |      |      |
|----|----|------|------|
| 10 | 11 | 1785 | 2180 |
|----|----|------|------|

## APPENDIX B

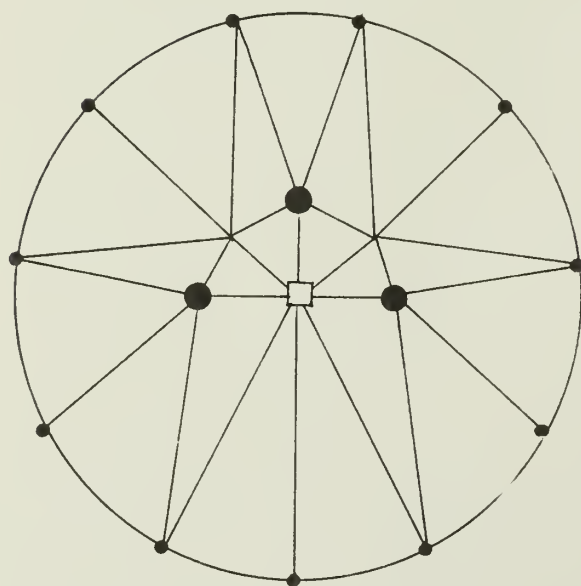
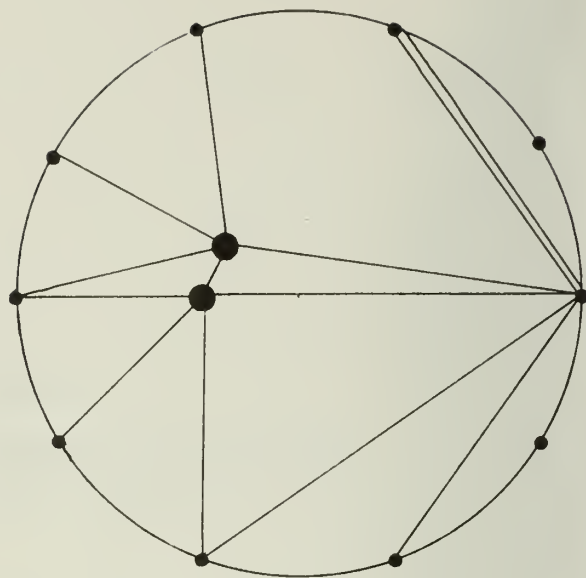
Four reducible configurations that require effective reducers with interior vertices.

Configuration

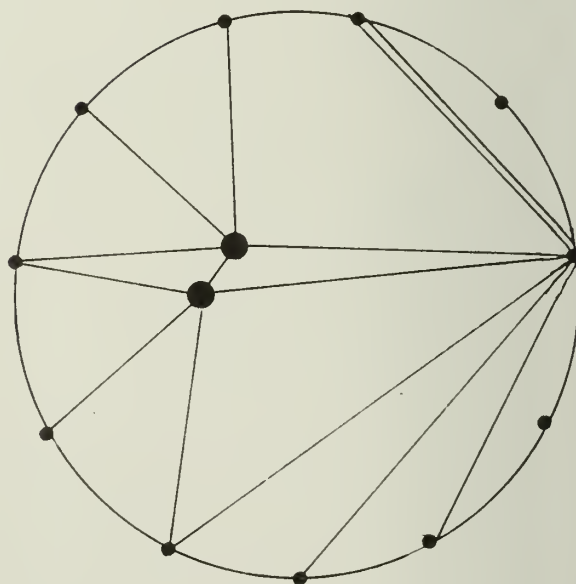


$m = 6$        $n = 10$   
 direct = 531  
 indirect = 672

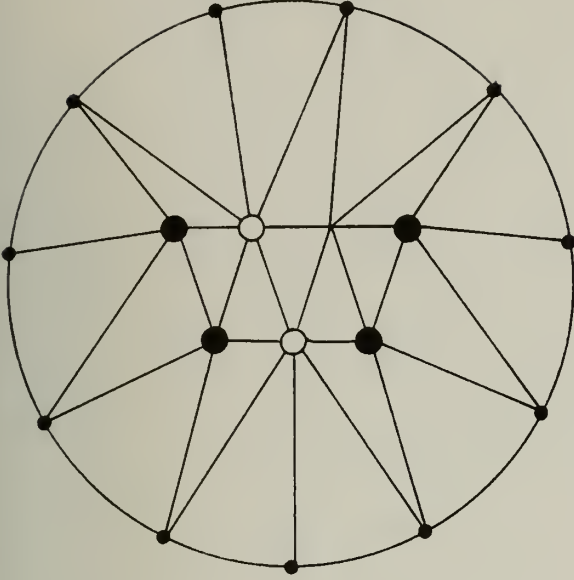
Reducer



$m = 6$        $n = 11$   
 direct = 1078  
 indirect = 3452

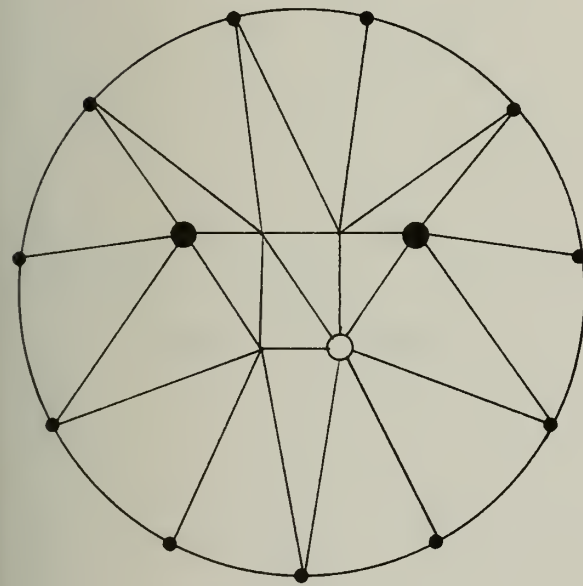
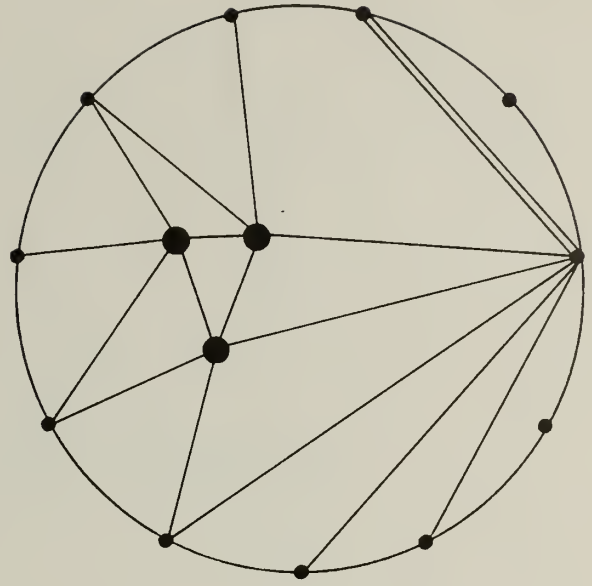


Configuration

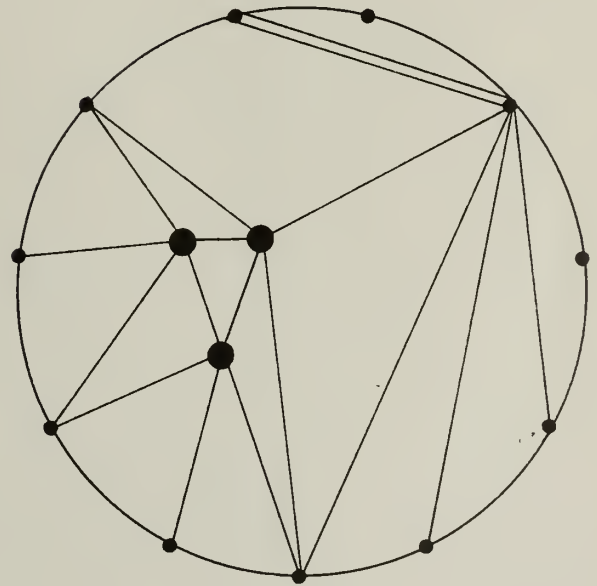


$m = 7$        $n = 11$   
 direct = 1297  
 indirect = 2796

Reducer



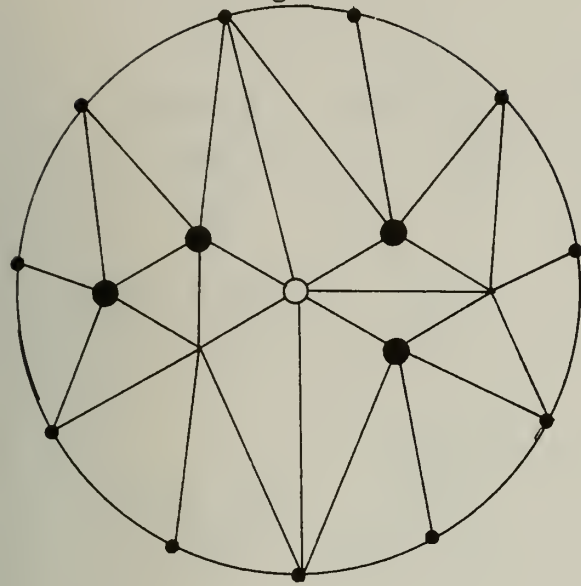
$m = 6$        $n = 11$   
 direct = 1109  
 indirect = 3309



## APPENDIX C

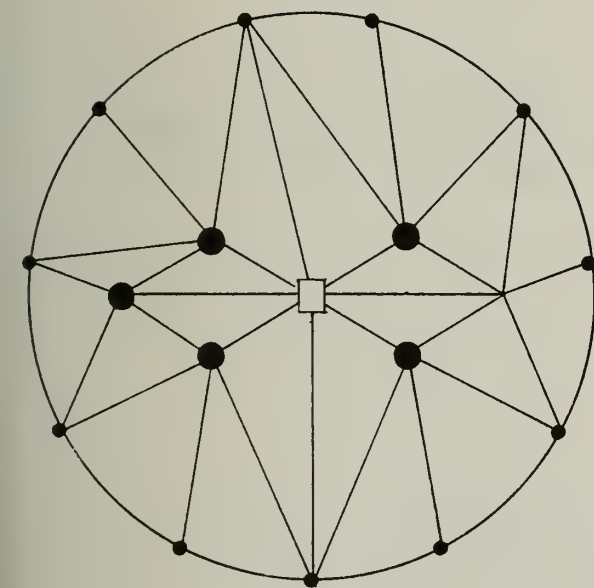
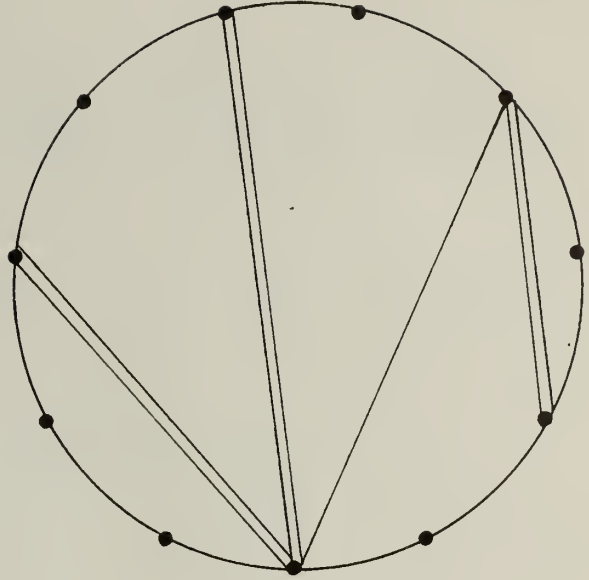
Two reducible 11-ring configurations and  
their effective reducers.

Configuration

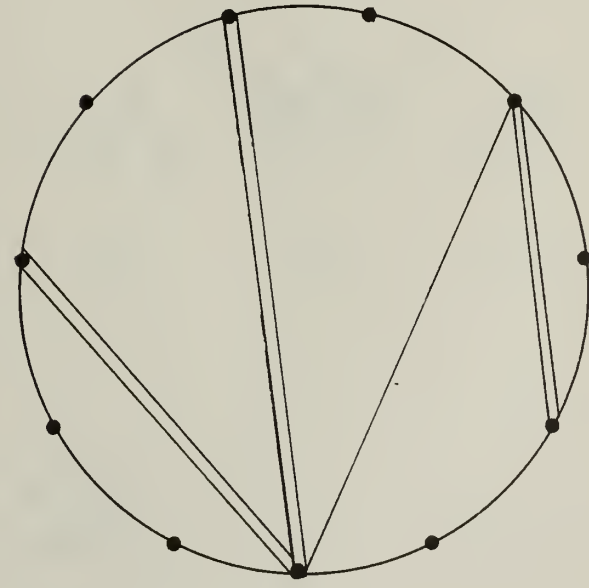


$m = 7$        $n = 11$   
 direct = 1322  
 indirect = 3680

Reducer



$m = 7$        $n = 11$   
 direct = 1287  
 indirect = 3855



## APPENDIX D

Irreducible 9-ring configurations.



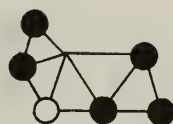
direct = 224  
indirect = 376



221  
376



224  
378



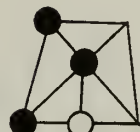
257  
315



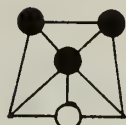
197  
456



direct = 219  
indirect = 362



224  
356



225  
352



201  
454



direct = 243  
indirect = 329



224  
443



223  
350



222  
336



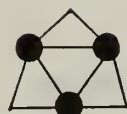
206  
484



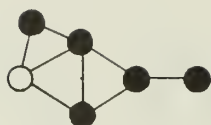
direct = 208  
indirect = 459



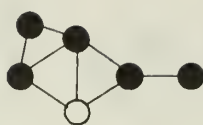
218  
332



212  
320



direct = 218  
indirect = 402



215  
392



237  
267



265  
326



direct = 160  
indirect = 654



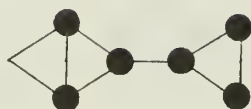
194  
480



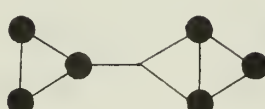
219  
410



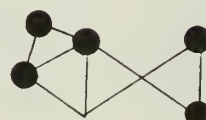
247  
312



direct = 235  
indirect = 293



236  
308



268  
301



219  
358



direct = 219  
indirect = 366



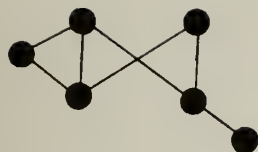
201  
470



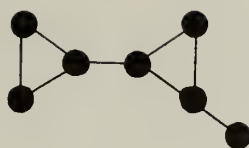
252  
275



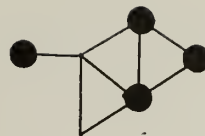
246  
281



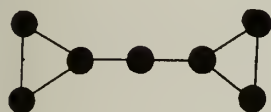
direct = 249  
indirect = 309



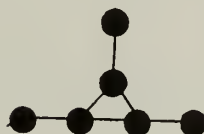
238  
332



216  
354



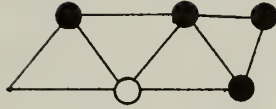
direct = 238  
indirect = 354



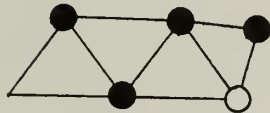
215  
440

## APPENDIX E

Two irreducible 9-ring configurations  
with additional good colorations.



direct colorations = 224  
 indirect colorations = 218  
 additional good colorations = 14



direct colorations = 219  
 indirect colorations = 239  
 additional good colorations = 18

## VITA

John Allen Koch was born in Urbana, Illinois, on October 31, 1948. He graduated from Bucknell University with a B.S. in Mathematics in May, 1970. He received his M.S. in Computer Science from the University of Illinois in May, 1972. He has been a Research Assistant in the Department of Computer Science from September, 1970, to May, 1976.

He was elected to the Mathematics honorary, Pi Mu Epsilon, in 1969, to the Society of Scientific Researchers, Sigma Xi, in 1972 and to the National Scholastic Honorary, Phi Kappa Phi, in 1975. He is a member of the American Mathematical Society and the Association for Computing Machinery.

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                   |                  |                                                |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|------------------|------------------------------------------------|
| BIBLIOGRAPHIC DATA<br>HEET                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 1. Report No.<br>UIUCDCS-R-76-802 | 2.               | 3. Recipient's Accession No.                   |
| Title and Subtitle<br><br>COMPUTATION OF FOUR COLOR IRREDUCIBILITY                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                   |                  | 5. Report Date<br>May 1976                     |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                   |                  | 6.                                             |
| Author(s)<br><br>John Allen Koch                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                   |                  | 8. Performing Organization Rept.<br>No.        |
| Performing Organization Name and Address<br><br>Department of Computer Science<br>University of Illinois at Urbana-Champaign<br>Urbana, Illinois 61801                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                                   |                  | 10. Project/Task/Work Unit No.                 |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                   |                  | 11. Contract/Grant No.<br><br>NSF MCS 73-03408 |
| Sponsoring Organization Name and Address<br><br>National Science Foundation<br>Washington, D.C.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                   |                  | 13. Type of Report & Period<br>Covered         |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                   |                  | 14.                                            |
| Supplementary Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                                   |                  |                                                |
| Abstracts<br><p>A critical part of a possible proof of the Four Color Conjecture would be the determination of the reducibility or irreducibility of a collection of configurations. A configuration is a planar graph bounded by a simple circuit (ring) of vertices. Informally, a reducible configuration is one that cannot be a subgraph of a minimal counterexample to the Four Color Conjecture. Formally, a configuration is irreducible when a certain collection of known methods of proving reducibility fail.</p> <p>D-reduction and C-reduction (methods of proving reducibility defined by H. Heesch in <u>Untersuchungen zum Vierfarbenproblem</u>) are used to determine reducibility or irreducibility of configurations with ring size <math>n</math>, <math>6 \leq n \leq 11</math>, and with fewer than <math>6-n</math> interior vertices. Two algorithms for performing D-reduction are presented. D-reduction was performed using all possible potential reducers. The methods devised to generate all possible potential reducers are discussed in detail. Four configurations that require reducers which contain interior vertices are shown.</p> |                                   |                  |                                                |
| Key Words and Document Analysis. 17a. Descriptors<br><br>Four-Color<br>reducible<br>graphs<br><br>Finally, many of the results, programs and data structures described in this thesis have application in the reduction of configurations with ring size larger than 11.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                                   |                  |                                                |
| Identifiers/Open-Ended Terms                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                                   |                  |                                                |
| COSATI Field/Group                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                   |                  |                                                |
| Availability Statement                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                                   |                  |                                                |
| 19. Security Class (This Report)<br>UNCLASSIFIED                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                   | 21. No. of Pages |                                                |
| 20. Security Class (This Page)<br>UNCLASSIFIED                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                                   | 22. Price        |                                                |



JUL 1 1976







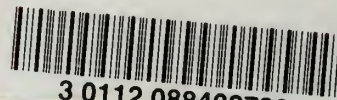




JUN 21 REC'D



UNIVERSITY OF ILLINOIS-URBANA  
510.84 IL6R no. C002 no. 800-805(1976  
Internal report /



3 0112 088402786